TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

VZDÁLENÉ OVLÁDÁNÍ PŘÍPRAVKU PRO PŘEDMĚT PMP

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Roman Halow**
Vedoucí práce: Ing. Tomáš Martinec, Ph.D.

Liberec 2014

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

REMOTE CONTROL OF THE EDUCATION BOARD FOR SUBJECT PMP

Diploma thesis

Study programme: N2612 – Electrotechnology and informatics
Study branch: 1802T007 – Information technology

Author: **Bc. Roman Halow**
Supervisor: Ing. Tomáš Martinec, Ph.D.

Liberec 2014

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: **2013/2014**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Roman Halow**
Osobní číslo:	**M11000260**
Studijní program:	**N2612 Elektrotechnika a informatika**
Studijní obor:	**Informační technologie**
Název tématu:	**Vzdálené ovládání přípravku pro předmět PMP**
Zadávající katedra:	**Ústav mechatroniky a technické informatiky**

Zásady pro vypracování:

1. Seznamte se s existujícím výukovým přípravkem pro předmět PMP.
2. Navrhněte takové úpravy nebo doplnění tohoto přípravku, které by umožnily jeho využití ve vzdálené laboratoři přes síť Internet.
3. Vyřešte zejména problém s programováním procesoru na dálku, se simulací stisknutí tlačítka, simulací natočení potenciometru a také přenos obrazové informace o stavu přípravku a obsahu displeje na přípravku.
4. Realizujte navržený hardware a software a otestujte upravený přípravek.

Rozsah grafických prací:	**dle potřeby dokumentace**
Rozsah pracovní zprávy:	**40–50 stran**
Forma zpracování diplomové práce:	**tištěná/elektronická**

Seznam odborné literatury:

[1] **BRTNÍK B., MATOUŠEK, D.: Programování mikrokontrolérů s jádrem 8051 v jazyce C. Praha: BEN, 2010, 152 stran, ISBN 978-80-7300-264-0**

[2] **Gook Michael: Hardwarová rozhraní - průvodce programátora. Praha: Computer Press, 2006, ISBN 80-251-1019-2**

[3] **Sosinsky Barrie: Mistrovství - počítačové sítě. Praha: Computer Press, 2010, ISBN 978-80-251-3363-7**

[4] **Dokumentace k použitému výukovému přípravku.**

Vedoucí diplomové práce:	**Ing. Tomáš Martinec, Ph.D.** Ústav mechatroniky a technické informatiky
Konzultant diplomové práce:	**Ing. Přemysl Svoboda** Ústav mechatroniky a technické informatiky
Datum zadání diplomové práce:	**10. října 2013**
Termín odevzdání diplomové práce:	**16. května 2014**

prof. Ing. Václav Kopecký, CSc.
děkan

L.S.

doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2013

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Abstrakt

Práce se zabývá vzdálenou správou výukového přípravku pro předmět PMP (Počítače a Mikropočítače). Popisuje funkční řešení, které bylo zrealizováno a otestováno v praxi. V závěru práce jsou shrnuty dosažené výsledky a návrh na vylepšení.

Během práce byly vytvořeny dvě aplikace v jazyce JAVA. První umožňuje programování mikroprocesoru AT89C51CC03, druhá pak slouží k ovládání navrženého hardwaru z Raspberry Pi pomocí sběrnice SPI. Nedílnou součástí je vytvořené webové rozhraní, kterým lze celý přípravek ovládat a zároveň sledovat pomocí webové kamery.

Klíčová slova: počítače a mikropočítače, vzdálená správa, Raspberry Pi, programování mikroprocesorů ATMEL

Abstract

The work deals with a remote control of the education board for PMP subject (Computers and Microcomputers). It describes a functional solution that was realized and tested in practice. In conclusion of the work there are summarized achieved results and a proposal for improvements.

During the work there were created two applications in Java language. The first allows microprocessor AT89C51CC03 programming, the second is used to control proposed hardware from Raspberry Pi using the SPI bus. An integral part is a created web interface that can control the entire agent and simultaneously observe it by a webcam at the same time.

Keywords: computers and microcomputers, remote control, Raspberry Pi, ATMEL microprocessor programming

Poděkování

Děkuji především svému vedoucímu diplomové práce Ing. Tomášovi Martincovi, Ph.D za odbornou pomoc, obětavé vedení a další cenné rady. Dále bych chtěl poděkovat také mé rodině a přátelům za jejich trpělivost a neustálou podporu.

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Diplomová práce se zabývá realizací vzdáleného ovládání přípravku pro výuku předmětu Počítače a Mikropočítače. V práci jsou popsány takové úpravy přípravku, díky kterým je možné přípravek sledovat a ovládat jeho komponenty z webového prohlížeče.

Zadání práce vzniklo na základě požadavků projektu virtuálních laboratoří, který v současné době na Technické Univerzitě v Liberci probíhá. Součástí tohoto projektu bude několik vzdáleně ovládaných přípravků a laboratorních úloh. Virtuální laboratoř může obsahovat několik přípravků umístěných například v temné místnosti nebo uzavřených do skříně.

Tento projekt je zaměřen především na zjednodušení studia osobám se sníženou tělesnou pohyblivostí a studentům kombinovaného studia tím, že nebude nutné, aby kvůli některým úlohám docházeli do laboratoří. Virtuální laboratoře by měly být dostupné také ostatním studentům.

První část práce obsahuje teoretický úvod, který objasní funkce a možnosti technologií použitých k realizaci. Popisuje samotný výukový přípravek společně s jeho komponenty. Zabývá se technologiemi pro programování mikroprocesorů a pro tvorbu webových aplikací. V dalších částech práce je vysvětlený základní princip vzdáleného ovládání, včetně postupu realizace všech potřebných součástí.

Mezi hlavní součásti lze zařadit software pro naprogramování mikroprocesoru AT89C51CC03, webové rozhraní pro vzdálenou správu a vytvořený hardware, který je umístěn přímo na přípravku.

Práce se dále zabývá zpracováním a streamováním videa, na kterém je možné přípravek sledovat, a k tomu potřebného osvětlení. Protože nepostradatelnou součástí funkčního celku je kompaktní počítač Raspberry Pi, obsahuje práce jeho základní popis včetně postupů, jak jej použít k realizaci vzdáleného ovládání.

Poslední část práce zahrnuje celkový postup, jak provést čistou instalaci na Raspberry Pi včetně vytvořeného softwaru, webové aplikace a ostatních podpůrných komponent. Protože čistá instalace může být poměrně náročná, zvláště pro uživatele, který nemá žádné zkušenosti s operačním systémem Linux, je postup velmi podrobný.

2 Popis výukového přípravku

Výukový přípravek slouží pro výuku předmětů zabývajících se programováním mikroprocesorů a používání hardwarových komponent. Jedním z předmětů, který je na tomto přípravku vyučován, je právě předmět PMP (Počítače a mikropočítače), na který se odkazuje i název práce.

Obrázek 2.1: Výukový přípravek

Základní vlastností přípravku je jeho rozdělení na odjímatelné moduly, které lze kdykoliv nahradit za jiné. V základní konfiguraci přípravek obsahuje 3 moduly: klávesnici, displej a modul s mikroprocesorem AT89C51CC03. Mikroprocesor je možné programovat pomocí sériové linky RS232, která je díky obvodu FT232RL virtualizována do sběrnice USB. Po překonfigurování jumperů lze použít i standardní sériovou linku bez vritualizace. Modul s mikroprocesorem dále obsahuje dva konektory pro sběrnici CAN a tlačítka RESET a PSEN. Tlačítko RESET slouží k restartování mikroprocesoru (nikoliv celého přípravku) a v kombinaci s tlačítkem PSEN také k přepnutí mikroprocesoru do programovacího módu. Tlačítko PSEN samostatnou funkcí nemá.

Součástí základní desky jsou další komponenty vhodné pro výuku. Jedná se o RTC obvod, LEDbar s posuvným registrem, piezoměnič, tepelné čidlo, tři LED diody, tlačítka a potenciometr. Základní deska je napájena pomocí sběrnice USB, popřípadě lze připojit i samostatný zdroj napájení. Kvůli komponentám, které vyžadují napěťovou úroveň 3,3 V je přípravek vybaven regulátorem napětí LM3940 [4]. Kompletní schéma přípravku lze nalézt v příloze C.

3 Použité technologie

Programovací jazyk Java

Programovací jazyk Java je objektově orientovaný jazyk, který se svou syntaxí podobá například programovacímu jazyku C#. Protože se jedná o jazyk interpretovaný, lze v něm napsaný software spustit na jakékoliv platformě, na které běží tzv. virtuální stroj (interpret) Javy. Protože při každém spuštění aplikace dochází k překladu tzv. mezikódu do strojového kódu pro konkrétní platformu, může být start těchto aplikací pomalejší.

Právě díky skutečnosti, že lze software napsaný v jazyce Java spouštět na jakékoliv platformě, byl tento jazyk zvolen pro vytvoření většiny softwaru v této práci.

PHP

PHP slouží převážně k programování dynamických webových stránek. Jedná se o skriptovací dynamicky typovaný jazyk jehož výstupem je obvykle HTML kód zobrazovaný u koncového uživatele. Skript je tedy vykonáván na straně serveru. Podporuje celou řadu komunikačních protokolů a disponuje velkým množstvím knihoven pro zpracování například grafiky nebo jiných médií. Další jeho výhodou je možnost běhu na různých platformách bez omezení běžně používaných funkcí.

V kombinaci s webovým serverem Apache a databází MySQL se PHP stal nejpoužívanějším skriptovacím jazykem pro tvorbu webových aplikací vůbec.

AJAX

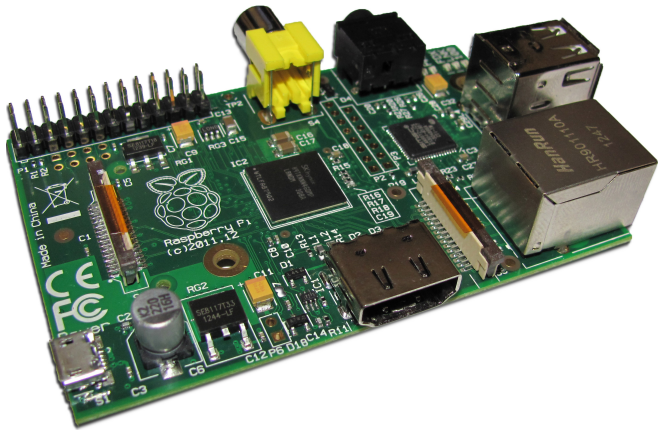
AJAX je obecné označení technologií, které umožňují provádět akce na webových stránkách bez nutnosti jejich opětovného načítání. Hlavní vlastností je možnost komunikace a přenos dat mezi serverem a klientem na pozadí. K přenášení souborů slouží objekt XMLHttpRequest. Tato technologie je často používaná u složitějších webových aplikací, například emailových klientů. Zkratka AJAX vychází ze slovního spojení Asynchronous JavaScript and XML.

JavaScript

JavaScript je programovací jazyk, který slouží k tvorbě webových aplikací, jedná se o tzv. klient skript. Je prováděn na straně klienta, který webovou stránku zobrazí. Nevýhodou může být možnost JavaScript vypnout ve webovém prohlížeči, což může způsobit absenci některých funkcí webu.

Kompaktní počítač Raspberry Pi

Raspberry Pi je kompaktní počítač postavený na architektuře ARM. Procesor pracuje s taktem 700 MHz, je však možné jej přetaktovat až na 1 GHz. Raspberry Pi verze B disponuje 512 MB operační paměti, což je dostatečná kapacita pro běh různých linuxových distribucí přeložených pro architekturu ARM. Celé zařízení je vidět na obrázku 3.1.



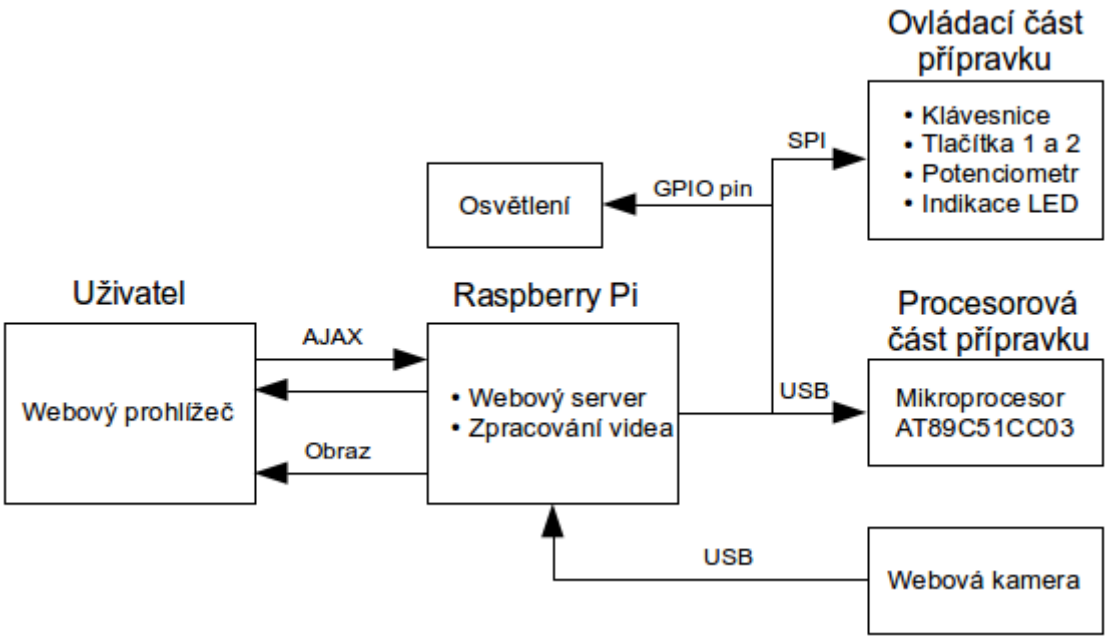
Obrázek 3.1: Kompaktní počítač RaspberryPi

Součástí Raspberry Pi není žádné paměťové medium, má pouze slot pro paměťové karty typu SD do maximální kapacity 32 GB. Další možností je připojení USB flash disku. Ostatní klíčové vlastnosti Raspberry Pi jsou uvedeny v následujícím seznamu [3]:

- HDMI výstup
- 2 USB porty
- Dostatek IO portů
- I2C a SPI sběrnice
- Seriová linka
- Ethernet port

4 Základní princip vzdáleného ovládání

Výukový přípravek je snímán kamerou a ovládán z webového rozhraní, které je spuštěno na Raspberry Pi. Z webového rozhraní jsou ovládány stisky kláves a tlačítek, nastavení hodnoty potenciometru a samotné programování mikroprocesoru. Všechny akce jsou na upravené desce přípravku znázorněny LED diodami tak, aby je bylo možné pomocí kamery sledovat. Základní princip, na kterém vzdálené ovládání funguje, je znázorněn na obrázku 4.1.



Obrázek 4.1: Blokové schéma základního principu ovládání

Uživatel sleduje prostřednictvím webového prohlížeče přípravek, a zároveň má k dispozici kontrolní panel pro jeho ovládání. Při stisku klávesy dojde k přenosu dat pomocí AJAXu k webovému serveru, a ten vyvolá příslušnou akci na kontrolní desce přípravku. Raspberry Pi komunikuje s kontrolní deskou přípravku pomocí sériové sběrnice SPI, která je ovládána aplikací SPI Controller. Aplikace SPI Controller byla vytvořena jako součást této práce a je popsána v kapitole 6.4. Při programování mikroprocesoru na přípravku je binární soubor opět pomocí AJAXu přenesen na webový server, který následně spustí s příslušnými parametry aplikaci Programmer. Vývoj této aplikace je součástí práce a je podrobněji popsán v kapitole 5.

5 Aplikace pro programování přípravku

Aplikací pro zápis programu do paměti mikroprocesorů je mnoho. Nicméně nebyla nalezena žádná, která by vyhovovala potřebám práce, a zároveň byla kompatibilní s operačním systémem Linux a architekturou ARM. V případě aplikace dodávané výrobcem mikroprocesoru, která se jmenuje Atmel Flip, není problém přímá kompatibilita s Linuxem (rozhraní je napsáno v jazyce Java), ale podpora způsobu, jakým operační systém Linux přistupuje k virtualizované sériové lince.

Aplikace Atmel Flip předpokládá v operačním systému Linux cestu k sériové lince `/dev/tty*`. V případě virtualizované sériové linky přes USB je však standardní cesta `/dev/ttyUSB*`. Tento problém neřeší ani vytvoření symbolického odkazu se správnou cestou. K aplikaci Atmel Flip bohužel nejsou dostupné zdrojové kódy a celkově by bylo její použití velmi náročné.

Další oblíbenou aplikací pro zápis do paměti mikroprocesorů je DFU Programmer. Jedná se o aplikaci vyvíjenou právě pro operační systém Linux, která podporuje mnoho mikroprocesorů od výrobce Atmel. Bohužel mikroprocesor, který je použit na přípravku, podporován není. Podporovány jsou pouze ty procesory, které mají přímo integrovaný obvod pro virtualizaci sériové linky do USB a všechny mikroprocesory řady AVR. K této aplikaci jsou dostupné zdrojové kódy, ale jejich úprava by byla velmi složitá a znamenala by nutnost pochopení funkce celého zdrojového kódu, jehož vývoj trval pravděpodobně několik let.

Vzhledem k těmto skutečnostem bylo rozhodnuto, že je nutné vytvořit vlastní aplikaci pro zápis do flash paměti mikroprocesoru přesně na míru potřebám práce. Vývoji a popisu funkčnosti této aplikace jsou věnovány následující podkapitoly 5.1, 5.2, 5.3 a 5.4.

5.1 Popis souboru Intel HEX

Soubor Intel HEX slouží k ukládání binárních dat, obvykle reprezentujících obsah paměti daného zařízení. V případě mikroprocesorů se často jedná o přeložený program, tedy posloupnost instrukcí ukládaných do paměti (ROM, EEPROM, FLASH a dalších). K vysvětlení, jak funguje aplikace pro zápis programu do paměti mikroprocesoru, je nejprve nutné pochopení samotné struktury tohoto souboru.

Intel HEX je textový soubor obsahující ASCII znaky, kde každá dvojice reprezentuje čísla šestnáctkové soustavy a jednotlivé záznamy jsou od sebe rozlišovány odřádkováním. Záznamy vždy začínají znakem ":" a končí kontrolním součtem, který se skládá ze dvou znaků. Počet znaků na jeden záznam bez prvního znaku ":" je vždy sudý. Struktura dat je znázorněna v tabulce 5.1. Jeden řádek v souboru Intel HEX může vypadat například takto:

:10246200436455491420305`24F4649444500D64C33`

Značka ":"	Délka záznamu	Adresa	Typ	Data	Kontrolní součet
1-bajt	1-bajt	2-bajty	1-bajt	n-bajtů	1-bajt

Tabulka 5.1: Formát jednoho záznamu v souboru Intel HEX

Jednotlivé položky tabulky 5.1 jsou popsány v následujícím seznamu:

Značka: ":" označuje začátek záznamu

Délka záznamu: definuje délku záznamu

Adresa: adresa v paměti, od které mají být data ukládána

Typ: typ záznamu

Data: data záznamu

Kontrolní součet: slouží ke kontrole přenesených dat

Kontrolní součet záznamu je vypočítán jako dvojkový doplněk součtu dvojic hexadecimálních čísel. Výpočet je vyjádřen rovnicí 5.1.

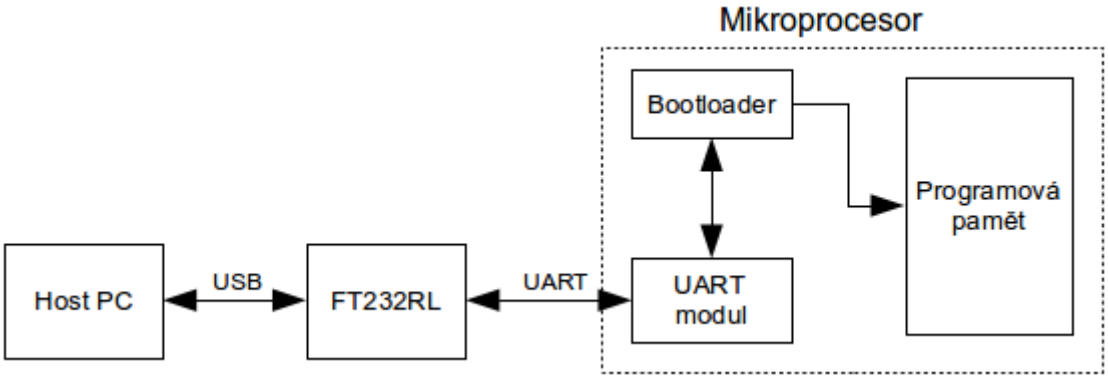
Příklad jednoho záznamu bez kontrolního součtu: :020000050F00

$$KS = 01h + NOT(02h + 00h + 00h + 05h + 0Fh + 00h) = EA \tag{5.1}$$

Výsledná podoba záznamu s kontrolním součtem: :020000050F00EA

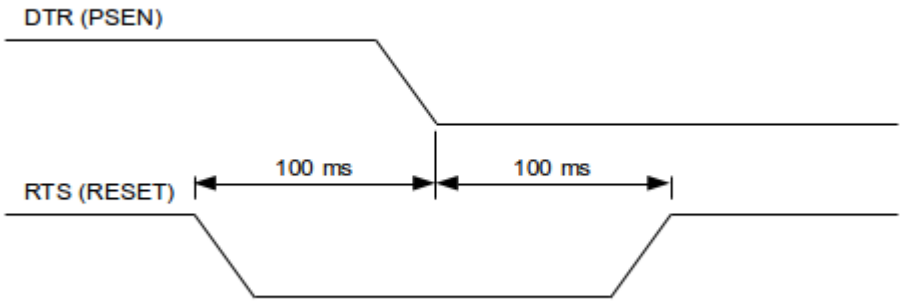
5.2 Komunikační protokol Atmel

Komunikační protokol Atmel využívá strukturu dat Intel HEX záznamů, která je popsána v kapitole 5.1. Jedná se o způsob komunikace mezi počítačem a bootloaderem mikroprocesoru. Blokové schéma propojení komunikační části mikroprocesoru s PC je na obrázku 5.1.



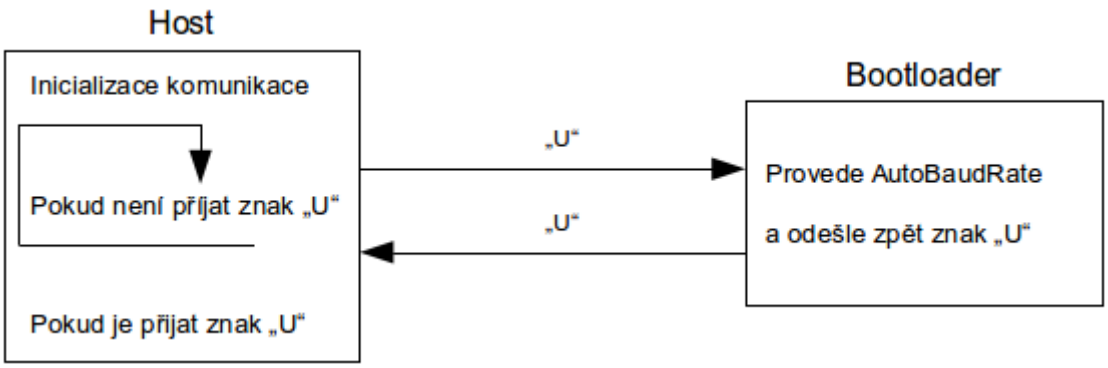
Obrázek 5.1: Blokové schéma komunikační části mikroprocesoru s PC

Před zahájením komunikace je nejprve zapotřebí uvést bootloader mikroprocesoru do programovacího režimu. Programovací režim je na přípravku možné aktivovat buď hardwarově pomocí tlačítek `RESET` a `PSEN`, nebo softwarově přes vodiče sériové linky `RTS` a `DTR`, kde vodič `RTS` odpovídá tlačítku `RESET` a `DTR` tlačítku `PSEN`. Časový diagram v jakém pořadí je zapotřebí stisknout tlačítka nebo převést signály na vodičích sériové linky do správné úrovně, je na obrázku 5.2.



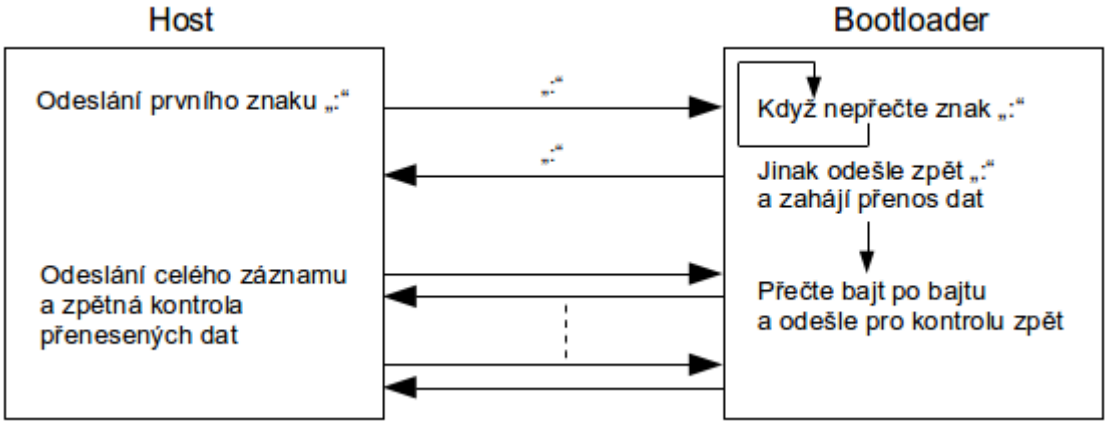
Obrázek 5.2: Časový diagram přepnutí bootloaderu do programovacího módu [2]

Bootloader na mikroprocesoru podporuje funkci AutoBaudRate, tato funkce umožňuje automatické nastavení rychlosti komunikace na základě rychlosti zařízení host (zařízení, které inicializuje komunikaci). K tomu, aby se AutoBaudRate provedl správně, je nejprve zapotřebí odesílat ze zařízení host znak `"U"`, a to až do doby, než dojde od bootloaderu k odpovědi v podobě stejného znaku. Při každé odpovědi bootloaderu (i při běžné komunikaci) jsou na konec vždy přidány řídící znaky `"\r\n"`. Postup, jakým probíhá AutoBaudRate, je na obrázku 5.3.



Obrázek 5.3: Blokové schéma provedení AutoBaudRate na Bootloaderu [1]

Po úspěšném provedení AutoBaudRate začne přenos vlastních dat. Data se posílají znak po znaku a bootloader každý přijmutý znak odešle pro kontrolu přenosu zpět. Jestliže kontrolní součet odeslaných dat odpovídá kontrolnímu součtu, který vypočítal bootloader, odešle zpět sekvenci znaků `".\r\n"`. Přenos dat je znázorněn na obrázku 5.4.



Obrázek 5.4: Blokové schéma přenosu dat mezi host PC a Bootloaderem [1]

Pomocí tohoto protokolu lze také přenášet k Bootloaderu na mikroprocesoru příkazy. Tyto příkazy je možné rozdělit do dvou skupin. První skupina slouží k ovládání operací nad programovou pamětí, například k jejímu celkovému vymazání, nebo čtení. Druhá skupina příkazů se pak zabývá mikroprocesorem, například nastavováním security levelu nebo čtení informací o výrobci. Tyto příkazy se mohou lišit podle daného typu mikroprocesoru, nicméně alespoň jejich základní část by měla být shodná. V tabulce 5.2 je seznam základních příkazů používaných u mikroprocesoru AT89C51CC03, který je osazen na přípravku.

Protože se ke komunikačnímu protokolu Atmel nepodařilo získat dokumentaci, je celý implementovaný postup programovaní mikroprocesoru odvozen z odposlechu komunikace mezi aplikací Atmel Flip a mikroprocesorem.

Příkaz:	Popis:
:0100000307F5	vymazání celé flash paměti
:020000030300F8	start programu s restartem (po naprogramování flash)
:03000003060000F4	zapíše bajt BSB (definuje, zda pamět obsahuje data)
:020000050000F9	přečte kód výrobce

Tabulka 5.2: Základní příkazy komunikačního protokolu Atmel

5.3 Struktura a funkčnost vytvořené aplikace

Úkolem aplikace je zápis souboru typu Intel HEX do paměti mikroprocesoru. Nejprve tedy bylo implementováno čtení a zpracování souboru typu Intel HEX, následně pak samotné odesílání dat. Jak již bylo naznačeno v kapitole 3, tato aplikace byla napsána v programovacím jazyce Java. Aplikaci je možné rozdělit do čtyř hlavních tříd.

Třída Programmer

Třída `Programmer` je hlavní třídou aplikace (Main class). Zpracovává vstupní informace od uživatele v podobě parametrů a následně volá požadované operace. V této třídě je také uložena jednoduchá nápověda, která se zobrazí v případě chybně zadaných vstupních parametrů.

Třída DataStruct

`DataStruc` je třída, kde každá její instance obsahuje jeden záznam souboru Intel HEX. Vstupní data jsou datového typu `String` a pomocí konstruktoru jsou rozdělena na adresu, která je datového typu `Integer` a samotná data, která jsou ponechána jako datový typ `String`. Třída dále obsahuje metody pro návrat adresy a dat.

Třída ProgramFile

Tato třída slouží k otevření a zpracování souboru Intel HEX. Celý soubor je čten po řádcích a ukládán do listu jako typ `DataStruct`. Po úspěšném přečtení celého souboru seřadí pomocí algoritmu BubbleSort obsah listu od nejmenší do největší adresy záznamu. Z dat v takto seřazeném listu sestaví pole řetězců o velikosti 256 znaků, ve kterých už nejsou ukládány adresy ze souboru Intel HEX, ale pouze samotná data.

Třída FlashProgrammer

Třída `FlashProgrammer` získá od třídy `ProgramFile` pole řetězců o velikosti 256 znaků. K jednotlivým řetězcům přidá informaci o délce záznamu, adresu v paměti, kam se mají data zapsat, typ záznamu a na konec vypočítaný kontrolní součet. Doplněním všech těchto informací data odpovídají formátu Intel HEX. Nakonec třída odešle jednotlivé řetězce pomocí sériové linky k bootloaderu mikroprocesoru. Při přenosu provádí zpětnou kontrolu dat a v případě nalezení jakékoliv chyby proces přeruší a informuje uživatele.

5.4 Ovládání vytvořené aplikace

Protože od začátku byla snaha vytvořit aplikaci tak, aby byla zároveň i samostatně funkčním celkem, slouží k jejímu ovládání jednoduché vstupní parametry. Tyto parametry lze zapisovat v libovolném pořadí a vždy začínají znaky `"--"`. Seznam všech parametrů včetně popisu funkce je v tabulce 5.3. Spuštění programu s příslušnými parametry je vidět na příkladu 5.1.

```
Programmer.jar --port /dev/ttyUSB0 --erase --program example.hex
--start
```

Příklad 5.1: Spuštění aplikace Programmer

Parametr:	Funkce:
--program	definuje cestu k binarnímu souboru programu (Intel HEX)
--port	definuje cestu k sériovému portu (např. /dev/ttyUSB0)
--memory	vybere paměť mikroprocesoru (eeprom, flash - vychozí)
--erase	vymaže celou paměť flash
--info	zobrazí informace o připojeném mikroprocesoru
--start	po naprogramovaní program automaticky spustí (s restartem)

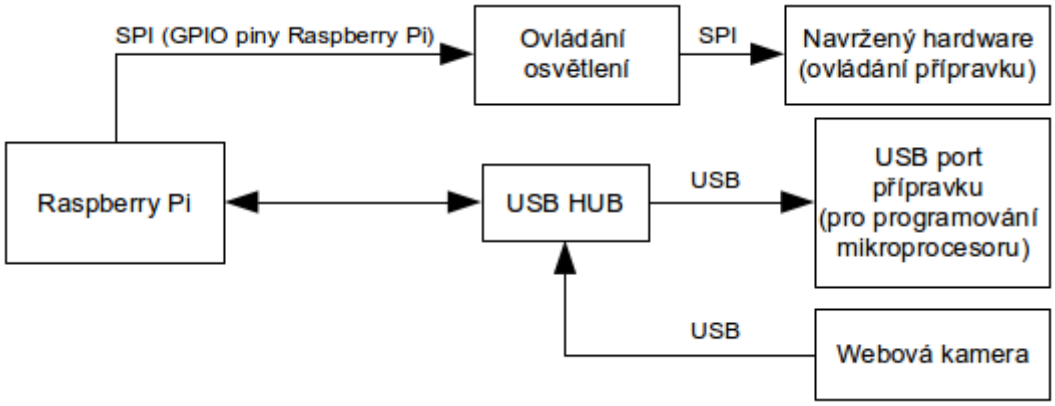
Tabulka 5.3: Vstupní paramtery aplikace pro naprogramovaní mikroprocesoru

Některé kompilátory zdrojového kódu pro mikroprocesory mají výstupní soubory s příponou `*.ihx`. Tyto soubory aplikací podporovány nejsou a je nejprve nutné provést převod do souboru s příponou `*.hex`. K převodu může být použita například jednoduchá konzolová aplikace Packihx, která je obsažena v balíčku kompilátoru SDCC.

Během provádění požadovaných operací jsou do terminálu vypisovány informace o aktuální činnosti aplikace. V případě zápisu do paměti flash je průběh zobrazován v procentech.

6 Ovládání prvků přípravku

Kapitola se zabývá návrhem hardwaru a softwaru, který je potřebný pro vzdálené ovládání prvků přípravku (potenciometru, klávesnice a tlačítek). Navržený a zrealizovaný hardware je popsán v kapitole 6.2. Software, který slouží k řízení komponent navrženého hardwaru, je blíže rozvinut v kapitole 6.4. Vzdálené ovládání lze provádět z webového rozhraní popsaného v samostatné kapitole číslo 9. Blokové schéma pospojování jednotlivých prvků ovládání je na obrázku 6.1.



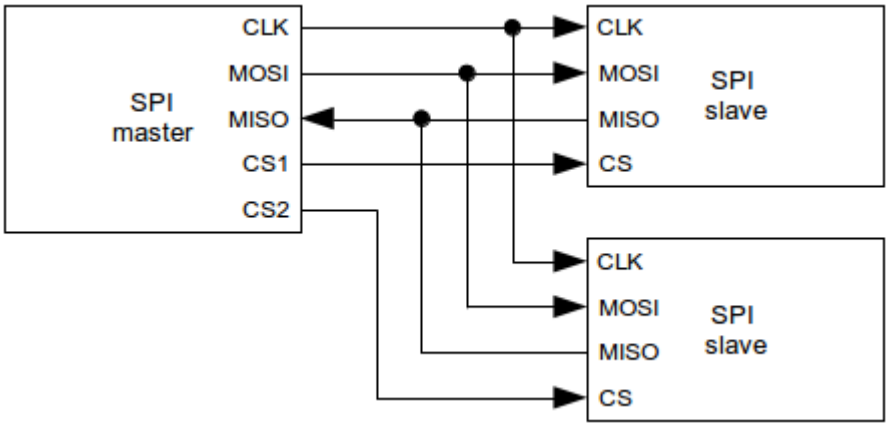
Obrázek 6.1: Blokové schéma pospojování jednotlivých prvků ovládání

6.1 SPI sběrnice

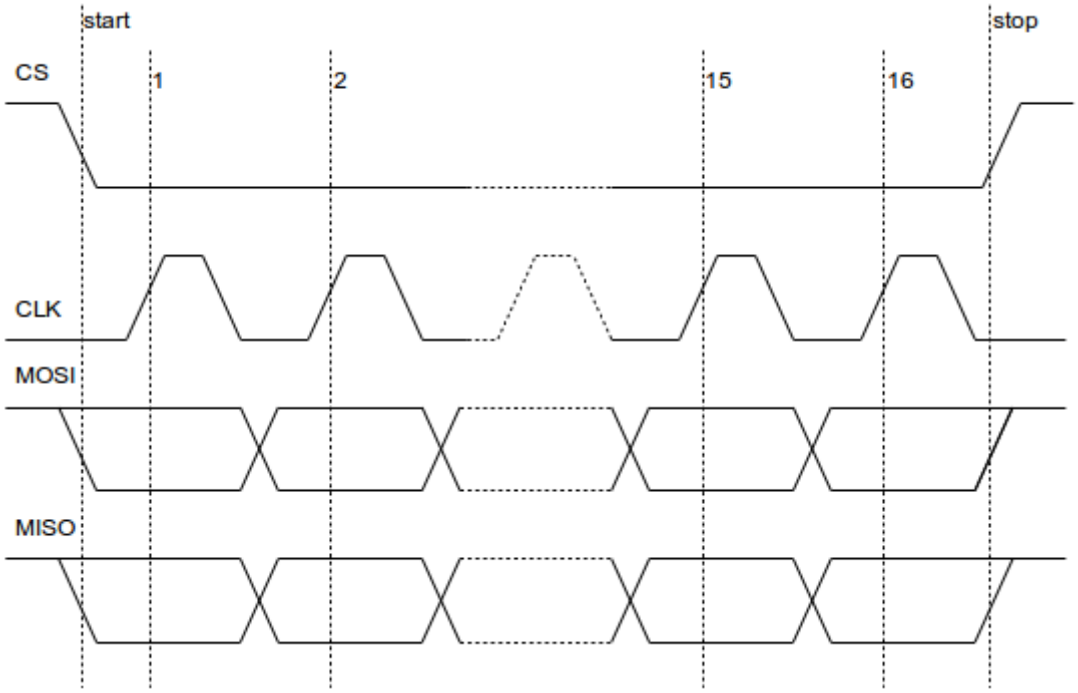
SPI sběrnice, z anglického Serial Peripheral Interface, slouží obvykle ke komunikaci mezi zařízeními na jednom plošném spoji, například mezi mikroprocesorem a D/A převodníkem. Jedná se o čtyř-vodičovou plně duplexní sběrnici (umožňuje komunikaci oběma směry současně). Adresace je prováděna pomocí tzv. ChipSelectu (občas deklarován jako SlaveSelect).

ChipSelect je samostatný vodič pro každé slave zařízení na sběrnici. Pokud chce master komunikovat s některým zařízením, musí jeho ChipSelect převést do dolní úrovně. Komunikaci ve směru master -> slave zajišťuje vodič MOSI, ve směru slave -> master pak MISO. Jakákoliv komunikace na sběrnici probíhá vždy na žádost

masteru. Blokové schéma zapojení SPI sběrnice je na obrázku 6.2. Časový diagram průběhu komunikace je na obrázku 6.3.



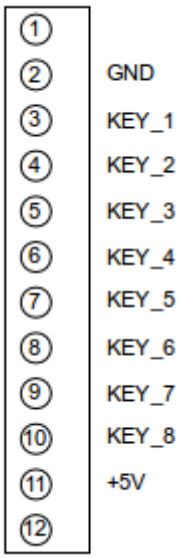
Obrázek 6.2: Základní schéma zapojení SPI sběrnice



Obrázek 6.3: Časový průběh komunikace po SPI sběrnici

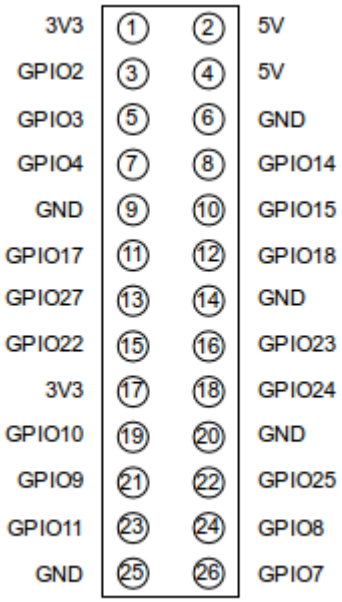
6.2 Navržený hardware pro ovládání přípravku

Navržený hardware pro ovládání přípravku je připojen na místo odstraněné klávesnice, která při vzdáleném ovládání není potřeba. Zapojení konektoru klávesnice je na obrázku 6.4. Hardware je dále pomocí několika vodičů připojen na GPIO piny kompaktního počítače Raspberry Pi. Zapojení konektoru Raspberry Pi je na obrázku 6.5.



Obrázek 6.4: Zapojení konektoru klávesnice na přípravku

Na přípravku je umístěna maticová klávesnice 4x4. Ta byla v hardwaru pro vzdálené ovládání nahrazena dvěma obvody MAX395. Ty slouží jako spínače mezi jednotlivými piny klávesnice a z pohledu výukového přípravku tak nahrazují stisknutí konkrétní klávesy. Každý stisk klávesy rozsvěcuje příslušnou LED diodu tak, aby měl uživatel vizuální kontrolu, že ke stisku opravdu došlo. Rozsvěcení LED diod zajišťuje obvod TLC5925. Ten v případě potřeby připojí některý ze svých výstupů na zem, díky čemuž začne protékat LED diodou proud a ta se rozsvítí. Oba obvody, jak MAX395, tak TLC5925, jsou řízeny pomocí sběrnice SPI. Na stejném principu jsou také ovládána obě tlačítka umístěná na přípravku.

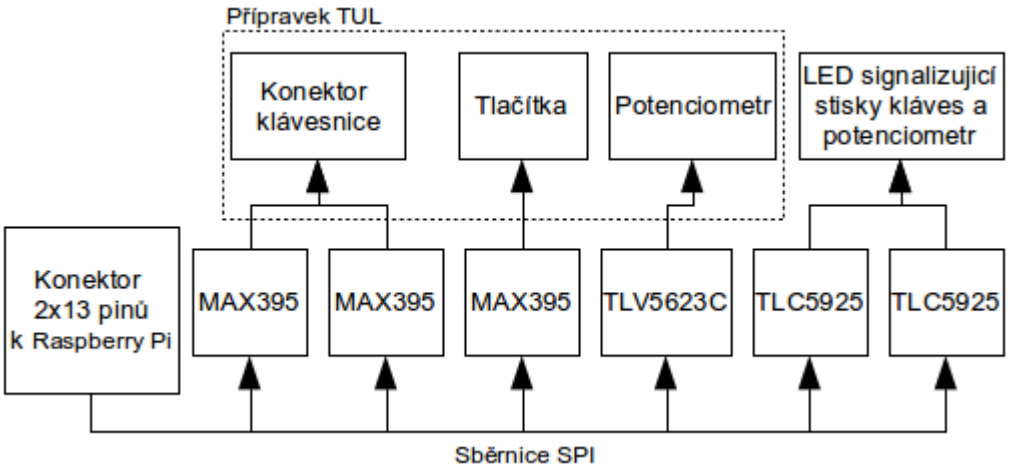


Obrázek 6.5: Zapojení konektoru GPIO pinů Raspbery Pi

V případě obvodu TLC5925 došlo během vývoje plošného spoje k omylu. V dokumentaci obvodu je uvedeno, že piny jsou konstantními zdroji proudu, což bylo mylně interpretováno jako fakt, že jsou zdrojem jak proudu, tak i napětí. Po ozkoušení bylo zjištěno, že výstupy obvodu jsou při sepnutí pouze připojovány k zemi s proudovým omezením, které je nastaveno referenčním odporem. Tento problém byl v zapojení později vyřešen přivedením kladného napětí k pinům obvodu, a tedy vytvoření obrácené logiky. Tento problém se vztahuje ke všem obvodům TLC5925, které byly v zapojení použity.

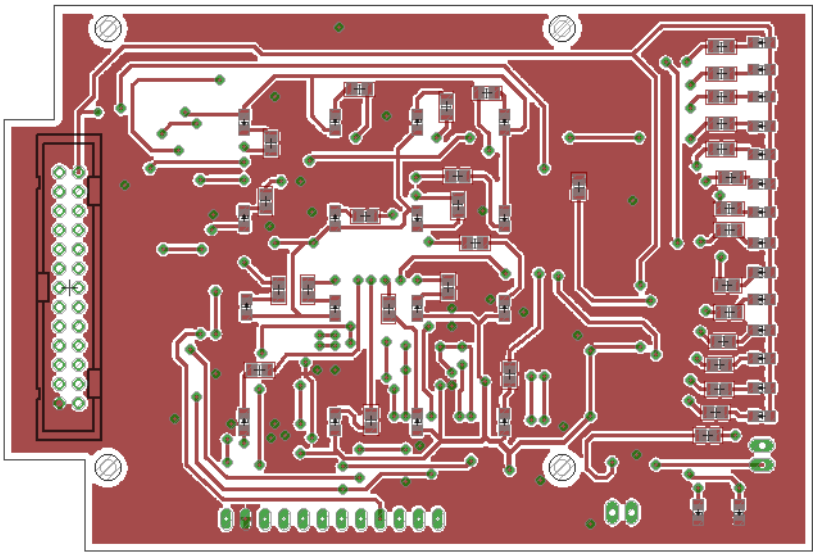
Potenciometr byl nahrazen D/A převodníkem TLV5623C, který má jako vstupní referenční napětí přivedeno napětí z nahrazeného potenciometru. Tím je vždy zajištěn stejný rozsah výstupního napětí, jaký byl u původního potenciometru. Vizuální kontrola pootočení potenciometru je zajištěna čtrnácti LED diodami a opět obvodem TLC5925. Rozsvícení všech diod odpovídá plnému natočení potenciometru, a tedy maximálnímu možnému výstupnímu napětí D/A převodníku. Obvod TLV5623C je také řízen pomocí SPI sběrnice. Blokové schéma zapojení jednotlivých komponentů navrženého hardwaru je na obrázku 6.6.

Všechna zařízení na SPI sběrnici jsou připojena k Raspberry Pi na piny k tomu určené, tedy MOSI a SCLK. Protože komunikace mezi Raspberry Pi a ostatními zařízeními na SPI sběrnici probíhá jednosměrně od Raspberry Pi, není pin MISO na Raspberry Pi, ani na integrovaných obvodech, připojen. ChipSelecty jednotlivých obvodů jsou připojeny každý na jeden GPIO pin Raspberry Pi.



Obrázek 6.6: Blokové schéma zapojení navrženého hardwaru

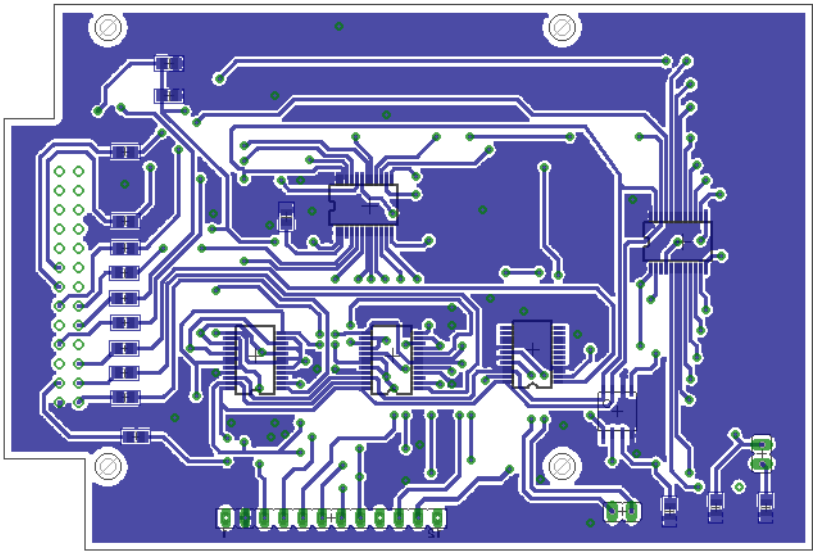
Celé zapojení obsahuje ještě mnoho podpůrných součástek (odpory, kondenzátory), jejichž použití vychází z dokumentace k daným obvodům, která je součástí DVD přílohy. Dále jsou mezi GPIO piny Raspberry Pi a obvody navržené desky sériově zapojeny odpory o velikosti 1 kΩ. Tyto odpory zajišťují kompatibilitu mezi napěťovou úrovní Raspberry Pi (3,3 V) a obvody použitými na desce (5 V). První verze schematu, která neobsahuje opravu zmiňovaného omylu v případě obvodu TLC5925, lze nalézt v příloze A, toto schéma bylo označeno za verzi číslo 1. Schéma, kde již je zanesena oprava, obsahuje příloha B a je označeno jako schéma verze 2. Návrh plošného spoje, odpovídající schématu verze 2, je na obrázku 6.7 a 6.8.



Obrázek 6.7: Navržený plošný spoj verze 2 - horní strana

Celý hardware byl navržen v aplikaci Eagle. Během návrhu bylo nutné pro tuto aplikaci vytvořit několik knihoven použitých součástek. Všechna vytvořená schémata a knihovny jsou součástí DVD přílohy.

Použité obvody byly vybrány tak, aby bylo možné je bez větších problémů zakoupit v České Republice a nebylo je tedy nutné v případě budoucí výroby objednávat v zahraničí. Dalším faktorem byl při výběru co nejnižší počet podpůrných součástek, které je potřeba při zapojení obvodu použít.



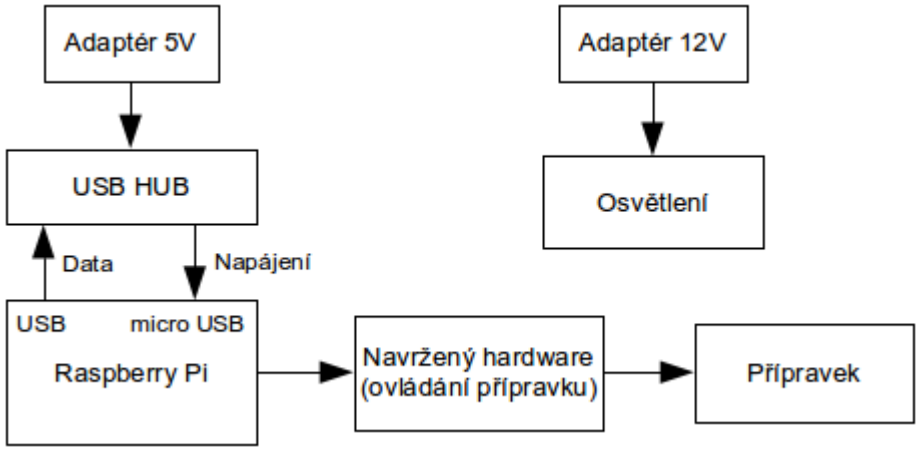
Obrázek 6.8: Navržený plošný spoj verze 2 - spodní strana

6.3 Princip napájení hardwaru

Napájení všech komponent hardwaru je zajištěno dvěma adaptéry. První adaptér s napětím 5 V disponuje konektorem jack o průměru 3,8 mm a napájí USB rozbočovač. Druhý adaptér s napětím 12 V je určen výhradně pro napájení osvětlení popsané v kapitole 8 a disponuje konektorem jack 5 mm. Napájení z tohoto adaptéru je od ostatních zařízení úplně odděleno.

Samotný přípravek je možné napájet jak z USB konektoru, tak z navrženého hardwaru popsaného v kapitole 6.2. Aby byl přípravek odpojen od napájení vždy, když je Raspberry Pi vypnuto, bylo u přípravku napájení pomocí USB konektoru odpojeno vyjmutím propojky, která je ve schématu přípravku (příloha C) označena jako `JP1`. Přípravek je tedy napájen výhradně z GPIO konektoru Raspberry Pi přes navržený hardware.

Raspberry Pi je připojeno pomocí datového USB konektoru k rozbočovači, nicméně napětí přes toto propojení nevyužívá, dokud není k Raspberry Pi připojeno jeho hlavní napájení přes mikro USB konektor. Aby nebylo pro hlavní napájení nutné použít další adaptér, byl použit propojovací kabel, který je z jedné strany připojen pomocí mikro USB konektoru k Raspberry Pi jako hlavní napájení a z druhé pomocí USB konektoru opět k rozbočovači. Princip celého napájení je vidět na obrázku 6.9.



Obrázek 6.9: Blokové schéma napájení jednotlivých prvků vzdáleného ovládání

6.4 Software pro kontrolu navrženého hardwaru

Pro ovládání navrženého hardwaru byla napsána aplikace, která umožňuje zapisovat do jednotlivých zařízení (obvodů) na SPI sběrnici. Tato aplikace je spouštěna na Raspberry Pi automaticky při startu systému a naslouchá na portu `4309`. Automatické spouštění je popsáno v kapitole 10.5.

Na port `4309` jsou z webového rozhraní přeposílány požadované hodnoty, které aplikace nejprve zpracuje a následně provede jejich zápis do jednotlivých SPI zařízení. Při každém zápisu jsou přepsány hodnoty všech zařízení bez ohledu na to, zda se změnily.

Všechny hodnoty jsou přepisovány kvůli zjednodušení a zároveň ošetření situace, kdy uživatel změní na webovém rozhraní více položek najednou. Například, pokud změní hodnotu potenciometru a následně stiskne virtuální klávesu přípravku, dojde i k zápisu nové hodnoty potenciometru.

Původně bylo pro komunikaci po SPI sběrnici zamýšleno využití modulu WiringPi, který obsahuje většina operačních systémů pro Raspberry Pi, včetně zde použitého Raspbianu. O celou komunikaci by se tak staral operační systém a příslušný modul by pouze dostával data k odeslání, podobně, jako je tomu v Linuxu u sériové linky. Od tohoto řešení bylo ustoupeno, protože WiringPi podporuje pouze dvě SPI zařízení na sběrnici a zapojení vzdáleného ovládání jich vyžaduje šest.

Jako nejlepší řešení pro komunikaci po SPI sběrnici bylo zvoleno použití Java knihovny Pi4J, která umožňuje přistupovat jednotlivě ke všem pinům Raspberry Pi. Díky této knihovně byly napsány vlastní funkce pro komunikaci po SPI sběrnici, které sice nedosahují takových rychlostí jako WiringPi, ale pro účely práce jsou naprosto dostačující. Nejdůležitější část funkce pro zápis na SPI sběrnici je vidět na příkladu 6.1.

```
for (int i=7;i >= 0;i--){
    clk.low();
    Thread.sleep(1);
    if(Integer.parseInt(String.valueOf(data_sw1.charAt(i)), 2)
    ==1)mosi.high();
    else mosi.low();
    clk.high();
    Thread.sleep(1);
}
```

Příklad 6.1: Nejdůležitější část funkce pro zápis na SPI sběrnici

Ukázka obsahuje jednoduchý `for` cyklus, který provádí zápis datového slova odzadu. Pomocí podmínky aplikované na každou část dat, je určeno, zda bude zapsána horní, nebo dolní úroveň. Dále je automaticky vytvářen hodinový signál střídáním napěťové úrovně na vodiči CLK. Mezi každou změnou úrovně hodinového signálu je vloženo zpoždění 1 ms. Z tohoto zpoždění byla přibližně odhadnuta frekvence komunikace na 500 Hz.

Aplikace se skládá ze tří hlavních tříd. Hlavní třídou (Main Class) je třída `Spi_controller`, která slouží ke zpracování vstupních parametrů a následnému volání příslušných funkcí. Dále z třídy `Keyboard`, která zajišťuje správu obvodů nahra-

zujících klávesnici, a ze třídy `Potentiometer`, která se stará o obvody nahrazující potenciometr.

Data jsou z webového rozhraní na socket odesílána jako řetězec znaků, který obsahuje jednotlivé položky oddělené mezerou v následujícím pořadí: nová hodnota potenciometru, stisknutá klávesa přípravku nebo tlačítka a délka stisku. Odeslaný řetězec tedy vypadá například takto: `60 B 500`.

Parametr délka stisku klávesy určuje, jak dlouho bude příslušný spínač v obvodu MAX395 propojen. Pro běžný stisk klávesy, který lze mikroprocesorem na přípravku bez problému zachytit, a zároveň je vidět rozsvícení LED diody na obrazu snímaného přípravku, postačuje čas 200 ms.

7 Obrazové snímání přípravku

Aby bylo možné dění na přípravku sledovat z webového rozhraní, bylo zapotřebí vyřešit problém se snímáním a následným zpracováním videa. Video je snímáno pomocí webové kamery, která je prostřednictvím USB portu připojena k Raspberry Pi. Video je nutné nejprve převést do komprimovaného formátu a následně streamovat.

Pro převádění videa v reálném čase bylo vyzkoušeno několik enkodérů (například FLV1 nebo h.264) vždy v kombinaci s aplikací VLC [5]. Bohužel ani jedna z těchto variant se neosvědčila, protože komprese videa těmito formáty je výpočetně velmi náročná a Raspberry Pi nedisponuje dostatečným výkonem. Nakonec bylo zvoleno použití streamu ve formátu mjpeg, který nemá tak vysoké nároky, a zároveň jeho kvalita dostačuje.

S formátem videa mjpeg velmi dobře pracuje konzolová aplikace pro operační systém Linux, která se jmenuje Motion. Tato aplikace je původně určena pro správu a zpracování výstupu z bezpečnostních kamer, nicméně disponuje všemi potřebnými funkcemi pro potřeby této práce. Aplikace například dokáže měnit rozlišení videa, otáčet jej nebo automaticky nastavovat jas a kontrast. V konfiguračním souboru aplikace je zapotřebí nastavit několik stěžejních parametrů. Jejich seznam je možné nalézt v tabulce 7.1. Ostatní parametry, které výchozí konfigurační soubor obsahuje, jsou dostačující. Konfigurační soubor byl uložen do složky `/usr/local/var/www/motion.conf`. Aplikace Motion je automaticky spouštěna při každém startu Raspberry Pi.

Parametr:	Hodnota:	Funkce:
width	400	šířka obrazu
height	288	výška obrazu
rotate	0	otočení
framerate	4	počet snímků za vteřinu
webcam_quality	100	úroveň komprese
webcam_port	8081	port video-streamu
daemon	on	spustí Motion v režimu procesu na pozadí

Tabulka 7.1: Základní parametry konfiguračního souboru aplikace Motion

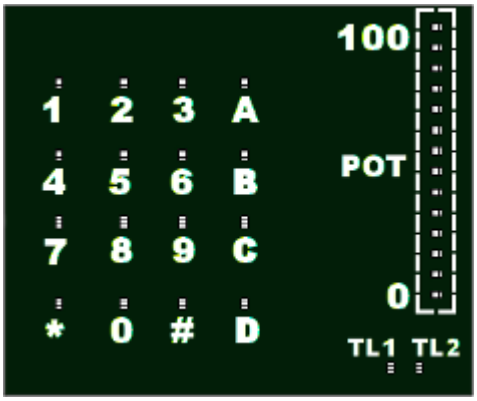
Zatížení procesoru Raspberry Pi je závislé na úrovni komprese videa, pokud je parametr `webcam_quality` nastaven na hodnotu 100, je komprese minimální a dochází tak ke snížení zatížení procesoru. Tento způsob, jak snížit nároky na výkon, má však za následek zvýšení datového toku videa. Při nastavení `webcam_quality` na hodnotu 100 a rozlišení 400 x 288 px se datový tok pohybuje okolo 3,5 Mbps.

Aby bylo zajištěno, že přípravek bude zaplňovat co největší část snímané plochy, byla kamera pootočena o 90 stupňů proti směru hodinových ručiček tak, že na výstupním videu z kamery se zdál být přípravek umístěn v horizontální poloze. Původně bylo video zpět o 90 stupňů otáčeno ihned v aplikaci Motion a výsledný video stream byl již pootočený správně. Bohužel tento proces je výpočetně velice náročný a při plné práci s webovým rozhraním již výkon Raspberry Pi nebyl dostačující. Z tohoto důvodu bylo zpětné otáčení videa přesunuto až na stranu webového prohlížeče, který video zobrazuje. Viz kapitola 9.2.

Do videa lze vložit jakýkoliv text, popřípadě aktuální datum a čas, jako tomu je ve výchozím nastavení. Protože zobrazovaná informace o čase obsahuje i setiny vteřiny, je vhodné ji využívat jako kontrolu, že přenos videa je v pořádku.

Aby bylo na zobrazovaném videu vzdáleného ovládání jasně rozpoznat, která LED dioda vyjadřuje stisk dané klávesy, byla vytvořena maska s popiskami, která je přiložena na tištěný spoj popisovaný v kapitole 6.2. Původně byla maska vytvořena kombinací bílého pozadí a černého textu, nicméně tato varianta nepříznivě ovlivňovala kvalitu snímaného obrazu, především pak rozdíl mezi rozsvícenou a zhasnutou LED diodou.

Jako nejlepší variantou byla nakonec zvolena kombinace tmavě zeleného pozadí a bílého textu. Díky podobnosti barev jak masky, tak celého přípravku kamera, lépe zvládá automatické nastavení jasu a rozsvícené LED diody s popiskami jsou velmi dobře viditelné. Výsledná podoba masky je na obrázku 7.1.



Obrázek 7.1: Maska s popiskami vzdáleného ovládání

7.1 Automatické spuštění aplikace Motion

K automatickému spouštění skriptu nebo aplikace po startu systému lze v Linuxu využít několik variant. Nejpoužívanější variantou je zápis do souboru `rc.local`, který se v použité linuxové distribuci Raspbian nachází ve složce `/etc`. Aby bylo možné Motion spustit jako proces na pozadí, je zapotřebí nejprve pomocí skriptu vytvořit složku `/var/run/motion`. Tato složka slouží jako místo pro uložení tzv. PID (proces ID) souborů a je vytvářena v případě, že aplikace bude mít více procesů. Obvykle si tyto složky vytvářejí jednotlivé aplikace automaticky, nicméně v případě Motion tato varianta na použité distribuci funkční není. Více informací o proměnném prostředí `/var/run` a obecně o hierarchii souborového systému v Linuxu by bylo mimo rozsah této práce. Do souboru `/etc/rc.local` je tedy pro automatické spuštění zapotřebí doplnit tyto dva řádky:

```
mkdir /var/run/motion/
motion -c /usr/local/var/www/motion.conf
```

Kde první řádek je příkaz pro vytvoření složky `/var/run/motion/` a druhý pro spuštění aplikace Motion s konfiguračním souborem uloženým v `/usr/local/var/www/motion.conf`.

Výhoda spouštění Motion v režimu procesu na pozadí je, že aplikace se poté lépe vypořádá s případnými problémy bez zásahu uživatele, například s odpojením a připojením kamery.

8 Osvětlení přípravku

Jak již bylo zmiňováno v úvodu práce, předpokládá se, že přípravek bude umístěn v temném prostředí. Výhodou je především potlačení okolního osvětlení a zlepšení kvality snímaného obrazu.

Za tímto účelem bylo navrženo osvětlení, které je součástí celé sestavy, viz příloha E. Skládá se ze dvou svítidel, přičemž každé osvětluje přípravek z jiného úhlu tak, aby byly co nejvíce potlačeny odlesky, které se na snímaném obrazu projevují velmi nepříznivě. Svítidla jsou rozsvícena a zhasínána automaticky podle toho, zda se někdo na snímaný obraz dívá.

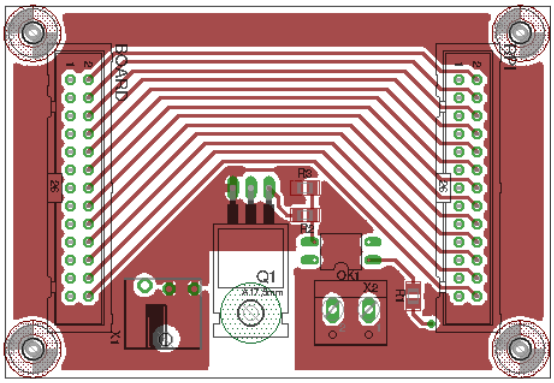
Každé svítidlo je sestaveno z LED pásku, který je umístěn do hliníkové lišty a je překrytý difuzorem. Ná délce LED pásku, která činí 160 mm je umístěno celkem devět samostatných LED diod. Svítivost jednotlivých LED výrobce neuvádí, nicméně bylo ozkoušeno, že je nad míru dostačující. Každý LED pásek je napájen napětím 12 V. Výsledná podoba svítidel je na obrázku 8.1.

Obrázek 8.1: Výsledná podoba svítidel

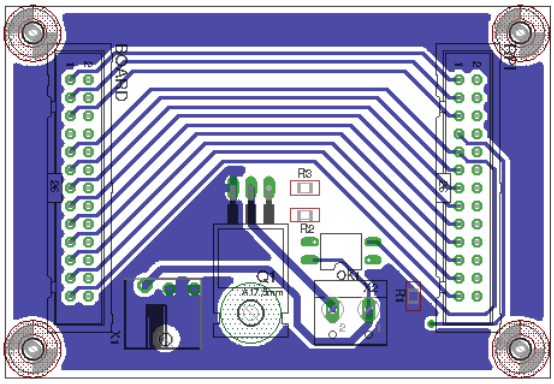
8.1 Navržený hardware pro ovládání osvětlení

Aby bylo možné ovládat osvětlení z Raspberry Pi, byl navržen hardware, který na základě stavu GPIO pinu spíná napětí 12 V. Hardware pro ovládání osvětlení a Raspberry Pi jsou vzájemně odděleny optickým členem.

Optický člen je na vstupu ovládán napětím 3,3 V GPIO pinu Raspberry Pi. Výstup optického členu spíná napětí 12 V, které je dále přivedeno na pin `gate` výkonového MOSFET tranzistoru IRFZ48. Návrh plošného spoje je vidět na obrázku 8.2 a 8.3. Kompletní zapojení je obsaženo v příloze D. Jako GPIO pin pro osvětlení byl zvolen pin číslo 7.



Obrázek 8.2: Navržený plošný spoj pro kontrolu osvětlení - horní strana



Obrázek 8.3: Navržený plošný spoj pro kontrolu osvětlení - spodní strana

Protože ostatní GPIO piny Raspberry Pi jsou použity pro ovládání přípravku, ale jsou zároveň součástí stejného konektoru jako pin číslo 7, bylo zapotřebí na tištěný spoj umístit dva konektory. Jeden vstupní a druhý výstupní, do kterého je dále připojen hardware pro ovládání přípravku. Tyto konektory jsou mezi sebou, s výjimkou pinu číslo 7, propojeny. V případě, že by byl hardware pro ovládání osvětlení vyjmut, je možné připojit konektor z hardwaru pro ovládání přípravku přímo k Raspberry Pi bez jakýkoliv úprav. Výsledná podoba navrženého hardwaru je na obrázku 8.4.

8.2 Skript pro ovládání osvětlení

Pro automatické rozsvěcení a zhasínání osvětlení byl napsán shell skript, který cyklicky kontroluje, zda je někdo připojen k portu, na kterém běží video stream

Obrázek 8.4: Hardware pro ovládání osvětlení

(port 8081). Pokud je někdo připojen, převede GPIO pin Raspberry Pi do horní úrovně a způsobí tak rozsvícení osvětlení, viz kapitola 8.1. Pokud nikdo připojen není, převede GPIO pin do dolní úrovně.

Pro kontrolu, zda je někdo k portu 8081 připojen, je využíván nástroj pro příkazový řádek, který se jmenuje Netstat. Jedná se o nástroj, který provede výpis všech navázaných příchozích i odchozích spojení. Z celého výpisu jsou následně příkazem `grep` vybrány pouze položky týkající se portu 8081.

K ovládání GPIO pinů na Raspberry Pi z shellu slouží příkaz `gpio`, kterým lze jak nastavovat režim GPIO pinů (vstup/výstup), tak na jednotlivé piny zapisovat. Celý skript je vidět v příkladu 8.1.

```
#!/bin/bash
sudo gpio mode 7 out
while true; do
    if netstat -t --numeric-ports | grep ':8081'; then
        sudo gpio write 7 1
    else
        sudo gpio write 7 0
    fi
    sleep 1
done
```

Příklad 8.1: Skript pro ovládání osvětlení

Skript je automaticky spouštěn po startu Raspberry Pi a běží tak neustále na pozadí. Způsob automatického spouštění již byl naznačen v kapitole 7.1 a je podrobněji popsán v kapitole 10.5. Soubor se skriptem byl nazván `osvetleni.sh` a je umístěn v kořenovém adresáři webového rozhraní.

9 Webový server

Chod webového rozhraní obstarává na Raspberry Pi nainstalovaný webový server Cherokee, který je obdobou známějšího webového serveru Apache. Webový server Chreokee byl zvolen proto, že jeho konfigurace je jednodušší, a především jí lze kompletně provádět z webového rozhraní. Není tedy zapotřebí ručně editovat žádné konfigurační soubory. Webový server je na Raspberry Pi spouštěn automaticky při každém startu.

9.1 Webové rozhraní pro ovládání přípravku

Jak již bylo naznačeno v předchozích kapitolách, ke vzdálenému ovládání přípravku bylo navrženo webové rozhraní, přes které je možné nahrávat do mikroprocesoru přípravku vytvořené aplikace, provádět stisky kláves na klávesnici, simulovat otáčení potenciometru a případně restartovat mikroprocesor přípravku. Všechny tyto úkony jsou zpětně signalizovány LED diodami.

Webové rozhraní využívá technologie HTML, CSS3, JavaScript, AJAX a PHP. Základním souborem je PHP skript `index.php`, který je při požadavku uživatele (otevření webové stránky) vykonán. Tento skript sestaví výslednou podobu webové stránky.

Skript `index.php` můžeme rozdělit do tří základních částí. První část sestaví hlavičku stránky, do které vloží soubor `style.css`. Soubor `style.css` slouží k upravě vizuální podoby webového rozhraní. Dále jsou pak vloženy soubory `script.js`, `read_log.js` a `picture_refresh.js`, které obsahují funkce v jazyce JavaScript pro zpracování požadavků uživatele.

Další část skriptu vloží do stránky cestu k mjpeg streamu s parametrem aktuální IP adresy, která je získána z proměnné webového serveru `$_SERVER['SERVER_ADDR']`. Více informací o vkládání videa na webovou stránku obsahuje kapitola 9.2. Poslední část skriptu provede vložení části pro ovládání přípravku uživatelem, tedy tlačítka pro simulaci stisku klávesnice, tlačítek přípravku, otáčení potenciometru a nahrávání programu do mikroprocesoru. Výsledná podoba webového rozhraní je na obrázku 9.1.

Obrázek 9.1: Vzdálené ovládání přípravku - webové rozhraní

K vyvolání požadované akce při stisku tlačítka webového rozhraní je použit v JavaScriptu tzv. EventListener. Pokud je stisknuto některé z tlačítek, EventListener tuto událost zachytí a zavolá funkci `button_number`. Zápis EventListeneru pro stisk tlačítka `A` je vidět na příkladu 9.1.

```
document.getElementById('A').addEventListener('click',button_number);}
```

Příklad 9.1: EventListener

Funkce `button_number` získá jako parametr identifikátor stisknutého tlačítka a následně si z webové stránky vyčte aktuální hodnotu potenciometru a délku stisku klávesy. Z těchto informací sestaví dotaz a odešle jej na server pomocí technologie AJAX metodou GET. Metoda GET slouží k přenosu informací od klienta (webového prohlížeče) k severu pomocí URL odkazu. Na následujícím příkladu je vidět sestavený dotaz GET:

```
client.open("GET","function.php?key="+this.id+"&pot="+document.
getElementById('pot-value').value+"&zpozdeni="+document.
getElementById('zpozdeni').value,true);
```

Příklad 9.2: Sestavený dotaz GET

Odeslaný dotaz je na serveru zpracován funkcemi v souboru `function.php`. Jak již bylo zmíněno v kapitole 6.4, po zpracování jsou data odeslána na port 4309, kde jsou přijata aplikací SPI Controller.

K nahrávání programu slouží formulář v kódu označený jako `uploadForm`. V tomto formuláři lze vybrat soubor formátu Intel HEX, který chceme do mikroprocesoru na přípravku nahrát. Jakmile dojde ke stisku tlačítka `Nahrát program`, je zavolána funkce upload, která vytvoří instanci třídy `FormData`. Třída `FormData` sestaví dotaz obsahující cestu k souboru Intel HEX a odešle jej metodou POST na server. Metoda POST slouží k přenášení dat na pozadí a na rozdíl od metody GET nejsou data vidět v adrese URL.

Po úspěšném přenesení souboru Intel HEX na server dojde k zavolaní funkce ze skriptu `upload.php`, která provede na základě přípony souboru opětovnou kontrolu, zda se opravdu jedná o soubor typu Intel HEX. Po úspěšné kontrole je soubor přenesen z dočasné složky serveru do složky `upload` a přejmenován na `file.hex`.

Pokud jsou všechny kroky vykonány v pořádku, provede skript `upload.php` spuštění programu `Programmer` s příslušnými parametry a dojde tak k naprogramování mikroprocesoru na přípravku. Ke spouštění programů na serveru přímo ze skriptu PHP slouží příkaz `exec`. Na příkladu 9.3 je vidět, jakým příkazem je v PHP aplikace `Programmer` spouštěna.

```
exec ("java -jar programmer/Programmer.jar"." "."--info"." "."--program".
" "."upload/file.hex"." "."--erase"." "."--start >> log.txt");
```

Příklad 9.3: Spuštění aplikace Programmer z PHP skriptu

Jako parametry jsou vkládány, kromě cesty k souboru Intel HEX, také parametry pro vymazání flash paměti mikroprocesoru a automatickému spuštění nahraného programu ihned po naprogramování (s restartem mikroprocesoru).

9.2 Zobrazení videa

K zobrazení videa snímaného přípravku na webovém rozhraní bylo původně zamýšleno využití volně dostupného Java appletu Cambozola, který umožňuje snadné zobrazení `mjpeg` streamu bez ohledu na to, zda je tento stream konkrétním prohlížečem podporován. Tento applet ovšem není digitálně podepsán a jeho spuštění vyžaduje schválení bezpečnostní výjimky. Protože u novějších verzí prostředí Java, které jsou vydávány od začátku roku 2014 je přidání této bezpečnostní výjimky velmi náročné, bylo od tohoto řešení nakonec úplně ustoupeno.

Jako další a zároveň finální řešení bylo zvoleno využití přímé podpory `mjpeg` streamu většinou prohlížečů bez jakýchkoliv doplňků. Jediným prohlížečem, který v současné době `mjpeg` stream nepodporuje je Internet Explorer.

Do webových stránek je video vloženo jako obyčejný obrázek pomocí tagu `img` pouze s tím rozdílem, že jako cesta k obrázku je uvedena IP adresa Raspberry Pi a port, na kterém se video stream nachází. Ná příkladu 9.4 je možné vidět způsob vložení videa.

```
<img src="http://<?php echo $_SERVER['SERVER_ADDR'] ?>:8081/video.
jpg" id="video" name="webcam" onload="InitialImage()">
```

Příklad 9.4: Vložení videa do webového rozhraní

Kde na místo `<?php echo $_SERVER['SERVER_ADDR'] ?>` je automaticky vložena aktuální IP adresa serveru (Raspberry Pi).

Bylo zjištěno, že některé prohlížeče po čase přestanou obraz video streamu aktualizovat. Tento jev je pravděpodobně způsoben snahou webových prohlížečů o úsporu systémových prostředků. Aby k tomuto jevu nedocházelo, byl napsán skript, který opakovaně způsobuje v prohlížeči načtení video streamu. Tento jev se nejvíce projevoval v prohlížeči Firefox.

Skript využívá v JavaScriptu metody `getTime()` a vlastnosti prohlížečů, že při každé změně adresy video streamu jej opětovně načtou. Návratová hodnota třídy `getTime()` je tedy přidávána za adresu video streamu a způsobuje tak neustálé opakování načítání. Celý skript je uložený v souboru `picture_refresh.js` v kořenovém adresáři webového rozhraní. Způsob přidávání návratové hodnoty metody `getTime()` je vidět na příkladu 9.5.

```
newImage.src = "http://<?php echo $_SERVER['SERVER_ADDR'] ?>:8081/
video.jpg?time=" + unique.getTime();
```

Příklad 9.5: Přidání návratové hodnoty metody getTime() k adrese video streamu

Jak již bylo naznačeno v kapitole 7, video stream je na výstupu z Raspberry Pi otočený o 90 stupňů proti směru hodinových ručiček. K otočení videa zpět bylo použito technologie CSS3 a vlastnosti `transform`. Ukázka stylu aplikovaného na zobrazované video je vidět na příkladu 9.6.

```
DIV#video-container{
    float: left;
    position: relative;
    -webkit-transform: rotate(90deg);
    -moz-transform: rotate(90deg);
    margin-top: 98px;
    margin-left: -44px;
}
```

Příklad 9.6: Ukázka CSS stylu aplikovaného na zobrazované video

9.3 Zobrazení výpisu aplikace Programmer

Součástí webového rozhraní je také výpis stavu aplikace `Programmer` během programování mikroprocesoru. Uživatel tak ví, v jaké fázi se programování nachází, popřípadě zda nedošlo k nějakým komplikacím.

Výpis aplikace `Programmer` je přesměrován do souboru `log.txt`, který je umístěn v kořenovém adresáři webového rozhraní. Obsah tohoto souboru je pak opakované přenášen pomoci AJAXu do webového rozhraní.

Pro čtení obsahu souboru `log.txt` byla napsána v jazyce JavaScript funkce, která se jmenuje `getFileFromServer` a jako vstupní parametr získává cestu k výše zmiňovanému souboru. Další částí skriptu je funkce, která využívá metody `setTimeout`. Tato metoda slouží k provádění dané akce po předem definovaném časovém intervalu, a díky ní je tedy možné funkci `getFileFromServer` volat v pravidelných intervalech. Doba intervalu byla nastavena na 210 ms.

Pokud se soubor `log.txt` nepodaří z nějakého důvodu načíst, je do webového rozhraní automaticky vložena hláška `"Chyba při čtení logu"`. Každý řádek odpovídá jedné provedené operaci aplikace Programmer. Průběh programování je zobrazován v procentech.

10 Postup čisté instalace na Raspberry Pi

Protože kompletní instalace všech potřebných komponent na Raspberry Pi může být poměrně náročná, obsahuje tato kapitola kompletní návod obsahující vše od instalace operačního systému na Raspberry Pi až po konfiguraci serveru Cherokee.

Součástí DVD přílohy je kompletní obraz paměťové karty, který je možné na paměťovou kartu zkopírovat pomocí příkazu `dd` popsaného v kapitole 10.1. Tento obraz obsahuje nainstalovaný operační systém Raspbian se všemi úpravami a potřebným softwarem. Slouží jak pro budoucí jednoduché instalace, tak pro lepší pochopení funkčního celku. Nainstalovaný systém, který je v přiloženém obrazu disku má následující přihlašovací údaje:

```
Uživatelské jméno: pi
Heslo: B0SR7Qd351
```

10.1 Instalace Raspbianu

Jako operační systém pro Raspberry Pi byl zvolen Raspbian. Jedná se o port distribuce Debian wheezy, která je přeložena pro architekturu ARM. Raspbian je ke stažení přímo na domovských stránkách výrobce Raspberry Pi v sekci downloads na adrese `http://www.raspberrypi.org/downloads`. Stažený soubor formátu `*.zip` nejprve rozbalíme. Výstupem by měl být soubor s příponou `*.img`.

Po úspěšném rozbalení je zapotřebí vložit do PC paměťovou kartu, na kterou bude Raspbian instalován. Kapacita karty by měla být alespoň 8 GB. Následující postup je popsán pro instalaci z OS Linux distribuce Debian, nicméně obecně platí, že je zapotřebí stažený obraz disku s Raspbianem přenést na paměťovou kartu. V OS Windows lze k tomuto účelu použít například Win32DiskImager a následující kroky pro překopírování obrazu na paměťovou kartu v OS Linux přeskočit.

Protože ve většině linuxových distribucí se vložené medium (paměťová karta) automaticky připojí, je nejprve nutné jí odpojit. K tomu potřebujeme znát cestu k paměťové kartě, kterou můžeme zjistit použitím příkazu `df -h`, který vypíše všechny připojené disky včetně paměťový karet. Postup je tedy následující:

```
df -h  #provede výpis všech připojených disků
umount /dev/sdb  #odpojí příslušný disk (paměťovou kartu)
```

Poté, co paměťovou kartu odpojíme, překopírujeme na ní stažený obraz disku (soubor s příponou `*.img`). K překopírování použijeme v linuxu příkaz `dd`:

```
dd bs=1M if=2013-12-20-wheezy-raspbian.img of=/dev/sdb
```

Kde parametr `if` je cesta k souboru s obrazem a `of` cesta k výstupnímu zařízení. Parametr `bs` vyjadřuje velikost bloků, po kterých se bude obraz kopírovat. Během kopírování není v terminálu viditelný žádný výstup a může trvat až desítky minut, je tedy zapotřebí setrvat. Po úspěšném zkopírování je v terminálu vypsáno několik informací, například průměrná rychlost kopírování.

Připravenou kartu můžeme vložit do Raspberry Pi a připojit napájení. Po startu se na obrazovce objeví modré okno pro základní konfiguraci. Jako první je potřeba rozšířit oddíl s Raspbianem přes celý disk. K tomu slouží položka nabídky `Expand Filesystem`. Roztažení oddílu proběhne automaticky bez dalšího zásahu uživatele. Další položka nabídky `Change User Password` slouží ke změně výchozího hesla uživatele `pi`. Protože na Raspberry Pi běží SSH server, je změna výchozího hesla důrazně doporučována. Poslední důležitou položkou nabídky je `Internationalisation Options`, která obsahuje podnabídky pro změnu časového pásma, rozložení klávesnice a výchozího jazyka systému.

10.2 Použití SSH

Aby nebylo k Raspberry Pi nutné mít stále připojený monitor a klávesnici, je na Raspberry Pi spuštěn SSH server. SSH je technologie, která umožňuje zabezpečené vzdálené připojení v textovém režimu jako konkrétní uživatel. V systému Windows je k připojení přes SSH možné použít aplikaci Putty. V Linuxu pak stačí obyčejný terminál a nainstalovaná aplikace SSH. Příkaz v terminálu pro připojení k SSH na Raspberry Pi z jiného počítače vypadá takto:

```
ssh 192.168.0.14 -l pi
```

Kde první parametr je IP adresa a druhý `-l pi` je uživatelské jméno, jakým se chceme přihlásit. Při prvním připojení dojde pravděpodobně k výzvě o potvrzení sdíleného klíče pro šifrovanou komunikaci. Po potvrzení klíče se objeví výzva k zadání hesla pro přihlašovaného uživatele, výchozí heslo je obvykle `raspberry`.

Pomocí technologie SCP (Secure Copy) lze na podobném principu jako funguje SSH přenášet mezi zařízeními soubory. Tato funkce je opět velmi užitečná už při samotné konfiguraci Raspberry Pi v textovém režimu. Následující příklad ukazuje použití příkazu `scp`:

```
scp spi_controller.jar pi@192.168.0.14:/usr/local/var/www/spi_controler
```

První parametr je soubor, který chceme přenést, druhý pak určuje, kam bude přenesen. Druhý parametr je složený z informace, jaký uživatel bude přihlášen (`pi`), zavináče, IP adresy počítače (Raspberry Pi), dvojtečky a cesty ke složce, do které bude soubor uložen.

10.3 Aktualizace OS Raspbian a firmwaru Raspberry Pi

Po čisté instalaci je téměř nutnost provést kompletní aktualizaci jak Raspbianu, tak samotného firmwaru pro Raspberry Pi. Aktualizaci Raspbianu provedeme sekvencí následujících příkazů:

```
sudo dpkg-reconfigure tzdata
sudo apt-get update
sudo apt-get upgrade
```

Před aktualizací samotného firmwaru je zapotřebí nainstalovat balíček `ca-certifikates` a `git-core`. Instalaci provedeme sekvencí příkazů:

```
sudo apt-get install ca-certificates
sudo apt-get install git-core
```

Po dokončení instalace stáhneme nový firmware a provedeme jeho aktualizaci:

```
sudo wget http://goo.gl/1BOfJ -O /usr/bin/rpi-update
sudo chmod +x /usr/bin/rpi-update
sudo rpi-update
```

Nakonec provedeme vypnutí Raspberry Pi, aby mohlo být znovu spuštěno s novým firmwarem a aktualizovanými balíčky:

```
sudo shutdown -r now
```

10.4 Instalace aplikace Motion

Aplikace pro streamování videa Motion je součástí standardních balíčků ve správci softwaru a její instalaci provedeme příkazem:

```
sudo apt-get install motion
```

Konfigurační soubor aplikace Motion, je uložen ve stejném adresáři jako webové rozhraní, které je součástí DVD přílohy. Rozbalení tohoto souboru je popsáno v kapitole 10.7.

Aby se aplikace Motion spouštěla automaticky po startu Raspberry Pi, je nutné přidat do souboru `/etc/rc.local` dva příkazy, jeden vytvoří po spuštění Raspberry Pi adresář `/var/run/motion` a druhý spustí aplikaci. Tyto dva příkazy lze do `rc.local` vložit sekvencí příkazů:

```
sudo echo "mkdir /var/run/motion" >>/etc/rc.local
sudo echo "motion -c /usr/local/var/www/motion.conf" >>/etc/rc.local
```

Po restartování Raspberry Pi by měla aplikace Motion startovat automaticky v režimu procesu na pozadí. Pro kontrolu, že proces Motion opravdu běží lze použít příkaz:

```
ps ax | grep motion
```

Výpis příkazu by měl obsahovat položku:

```
motion -c /usr/local/var/www/motion.conf
```

10.5 Automatické spouštění aplikací po startu

Automaticky po startu operačního systému je zapotřebí spouštět kromě aplikace Motion také skript pro osvětlení a aplikaci SPI Controller. Automatické spuštění skriptu pro osvětlení lze nastavit pomocí tohoto příkazu:

```
sudo echo "sh /usr/local/var/www/osvetleni.sh >> null &" >>
/etc/rc.local
```

Automatický start aplikace SPI Controller lze nastavit tímto příkazem:

```
sudo echo "java -jar /usr/local/var/www/spi_controller/
spi_controller_v2.jar &" >> /etc/rc.local
```

10.6 Server Cherokee a PHP

Protože webový server Cherokee není dostupný pro architekturu ARM jako softwarový balíček, je nutné jej nejprve zkompilovat. Stažení zdrojových souborů provede následující příkaz:

```
wget http://cherokee.pubcrawler.com/1.2/1.2.101/cherokee-1.2.101.tar.gz
```

V případě, že by se balíček nepodařilo z odkazu stáhnout, je dostupný také v DVD příloze této práce. V dalším kroku je zapotřebí balíček rozbalit, což lze provést příkazem:

```
tar -xfvz cherokee-1.2.101.tar.gz
```

Před kompilací je nutné nainstalovat balíček `gettext`, a až poté provést samotnou kompilaci rozbaleného balíčku. Oba tyto kroky provedeme pomocí sekvence těchto příkazů:

```
sudo apt-get install gettext
cd cherokee-1.2.101
./configure
make
sudo make install
sudo ldconfig
sudo checkinstall -D
sudo addgroup www-data
```

Po zadání příkazu `sudo checkinstall -D` bude nutné vyplnit popis (Description) instalovaného balíčku. Ostatní dotazy lze ponechat ve výchozím stavu a pouze je potvrdit klávesou Enter. Protože dosavadní postup nejenže nainstaluje server Chreokee, ale zároveň vytvoří balíček debianu, je možné instalaci v budoucnu provádět přes systém pro správu balíčků Raspbianu a předchozí kroky pro kompilaci tak přeskočit. Tento zkompilovaný balíček s verzí serveru Chreokee 1.2.101 je součástí DVD přílohy. Jeho instalaci lze provést příkazem:

```
sudo dpkg -i cherokee_1.2.101-1_armhf.deb
```

Aplikace pro zpracování PHP skriptů je dostupná jako balíček ve zdrojích softwaru, proto je instalace poměrně snadnou záležitostí a lze jí provést příkazem:

```
sudo apt-get install php5-cgi
```

U nainstalovaného PHP je před použitím nejprve nutné editovat konfigurační soubor a odstranit středník před parametrem `cgi.fix_pathinfo=1`. Tato operace povolí v PHP proměnnou PATH_INFO, čímž se zvýší kompatibilita s větším množstvím existujících PHP skriptů. Editaci lze provést terminálovým editorem `nano` tímto příkazem:

```
sudo nano /etc/php5/cgi/php.ini
```

10.7 Spuštění webové aplikace

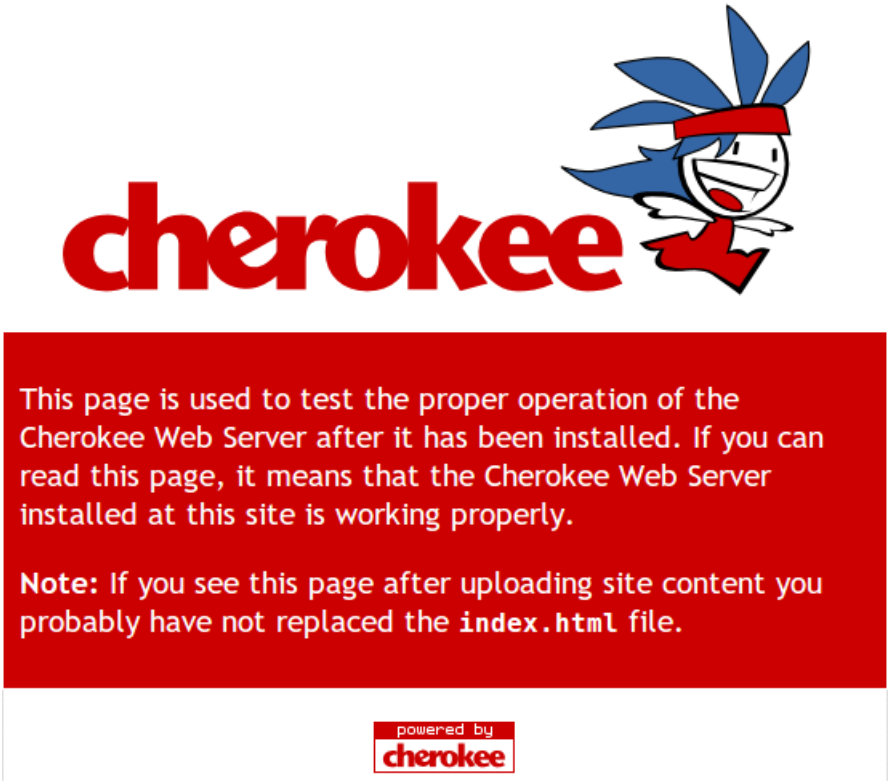
Pro úspěšné spuštění webové aplikace je nutné provést základní nastavení serveru Chreokee. K tomu slouží webové rozhraní, které je nutné před jeho použitím spustit. Spuštění webového rozhraní pro administraci provedeme příkazem:

```
sudo cherokee-admin -b
```

Výstupem tohoto příkazu je adresa, na které běží webové rozhraní pro administraci a aktuální přihlašovací údaje (jméno a heslo). Heslo je vždy platné pouze pro jednu relaci. Webové rozhraní administrace lze otevřít vložením IP adresy Raspberry Pi a portu 9090 do adresního řádku webového prohlížeče. Například `192.168.0.14:9090`.

Na úvodní stránce webového rozhraní je nutné webový server poprvé spustit zeleným tlačítkem `Start Server`. Pokud server běží správně, měla by být při vložení IP adresy Raspberry Pi do adresního řádku prohlížeče vidět stránka, jejíž náhled je na obrázku 10.1.

Na záložce `vServers` provedeme u výchozího serveru změnu položky `Directory Indexes` z `index.html` na `index.php`. Touto změnou zajistíme, že při požadavku na zobrazení stránky je vykonán skript index.php, který obsahuje webové rozhraní pro ovládání přípravku.



Obrázek 10.1: Kontrolní stránka webového serveru Cherokee

Součástí DVD přílohy je komprimovaný soubor s názvem `webove_rozhrani.tar.gz`, který obsahuje kompletní webové rozhraní pro ovládání přípravku, včetně souborů se skripty a potřebnými aplikacemi. Celý tento soubor je potřeba rozbalit do kořenového adresáře virtuálního serveru, který je ve výchozím nastavení serveru Cherokee `/usr/local/var/www`.

Soubor `webove_rozhrani.tar.gz` zkopírujeme do domovské složky uživatele `pi` na Raspberry Pi pomocí příkazu `SCP` a následně soubor rozbalíme do adresáře `/usr/local/var/www`. Tyto dva úkony lze provést sekvencí příkazů:

```
scp webove_rozhrani.tar.gz pi@192.168.0.14:/home/pi/
sudo tar -xvf webove_rozhrani.tar.gz -C /usr/local/var/www
```

Kde první příkaz slouží k překopírování souboru a je tedy prováděn v terminálu zdrojového počítače. Druhý příkaz pro dekompresi je pak nutné provést na Raspberry Pi.

11 Závěr

V rámci práce byl napsán vlastní software pro programování některých mikroprocesorů od společnosti Atmel, který je možné používat i jako samostatný celek. Tento software by ocenila především skupina studentů využívající k práci operační systém Linux, pro který není dostupný žádný vhodný software s těmito funkcemi, viz kapitola 5.

Výhodou celého řešení je, že uživateli stačí ke vzdálenému ovládání pouze webový prohlížeč s povoleným JavaScriptem. Ovládání tedy není vázáno používanou platformou a bylo vyzkoušeno také na zařízeních s operačním systémem Android. Úspěšně ozkoušeny byly také všechny nejběžněji používané webové prohlížeče s výjimkou Internet Exploreru, který jak již bylo zmíněno v kapitole 9.2 nepodporuje mjpeg stream.

Během testování se na displeji přípravku objevovaly odlesky osvětlení, které způsobovaly nečitelnost jeho pravé strany. Tento problém se nepodařilo vyřešit ani změnou úhlu osvětlení a nakonec bylo nutné část jednoho svítidla přelepit černou páskou (přibližně 20mm).

Softwarová část vzdáleného ovládání také řeší automaticky případné pády aplikací spuštěných na pozadí. Pokud k takovému pádu dojde, je aplikace opět spuštěna bez nutnosti restartování Raspberry Pi.

Hlavním cílem práce bylo zrealizování funkčního vzdáleného ovládání přípravku pro výuku předmětu Počítače a Mikropočítače. Tento cíl se podařilo splnit a funkční celek byl otestován v praxi. Příloha E obsahuje fotografii kompletního řešení v provozu.

Použitá literatura

[1] *In-System Programming - Serial Protocol Stack for C51 Products* [online]. [vid. 11. 12. 2013].
Dostupné z: http://www.atmel.com/Images/doc3a67133a3f86b.pdf

[2] ATMEL. *Datasheet AT89C51CC03* [online]. [vid. 5. 12. 2013].
Dostupné z: http://www.atmel.com/Images/doc4182.pdf

[3] *RaspberryPi FAQs* [online]. [vid. 5. 7. 2013].
Dostupné z: http://www.raspberrypi.org/faqs

[4] MARTINEC, Tomáš. *Dokumentace výukového přípravku*, 2009.

[5] BILIEN, J., DAOUD, A., STENAC, C., CELLERIER, A., SAMAN, J. [online]. *VideoLAN Streaming Howto*. [vid. 10. 9. 2013].
Dostupné z: http://www.videolan.org/doc/streaming-howto/en/

[6] SPELL, Brett. *Pro Java Programming*.
2nd ed. Berkeley, Calif.: Apress, 2005, xxiii, 694 p. ISBN 15-905-9474-6.

A Schéma vzdáleného ovládání verze 1

B Schéma vzdáleného ovládání verze 2

C Schéma výukového přípravku

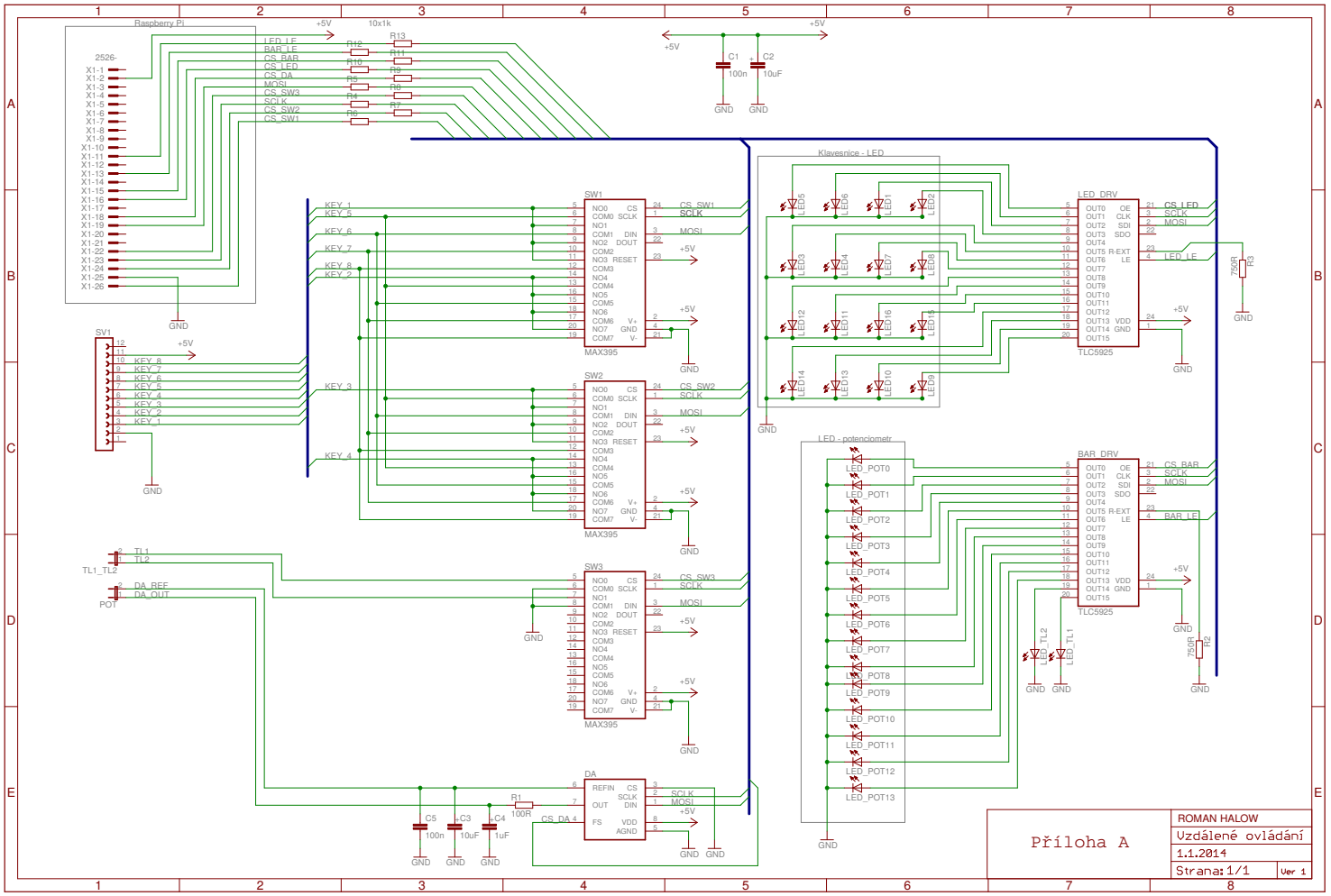
D Schéma ovládání osvětlení

E Fotografie zkompletovaného řešení

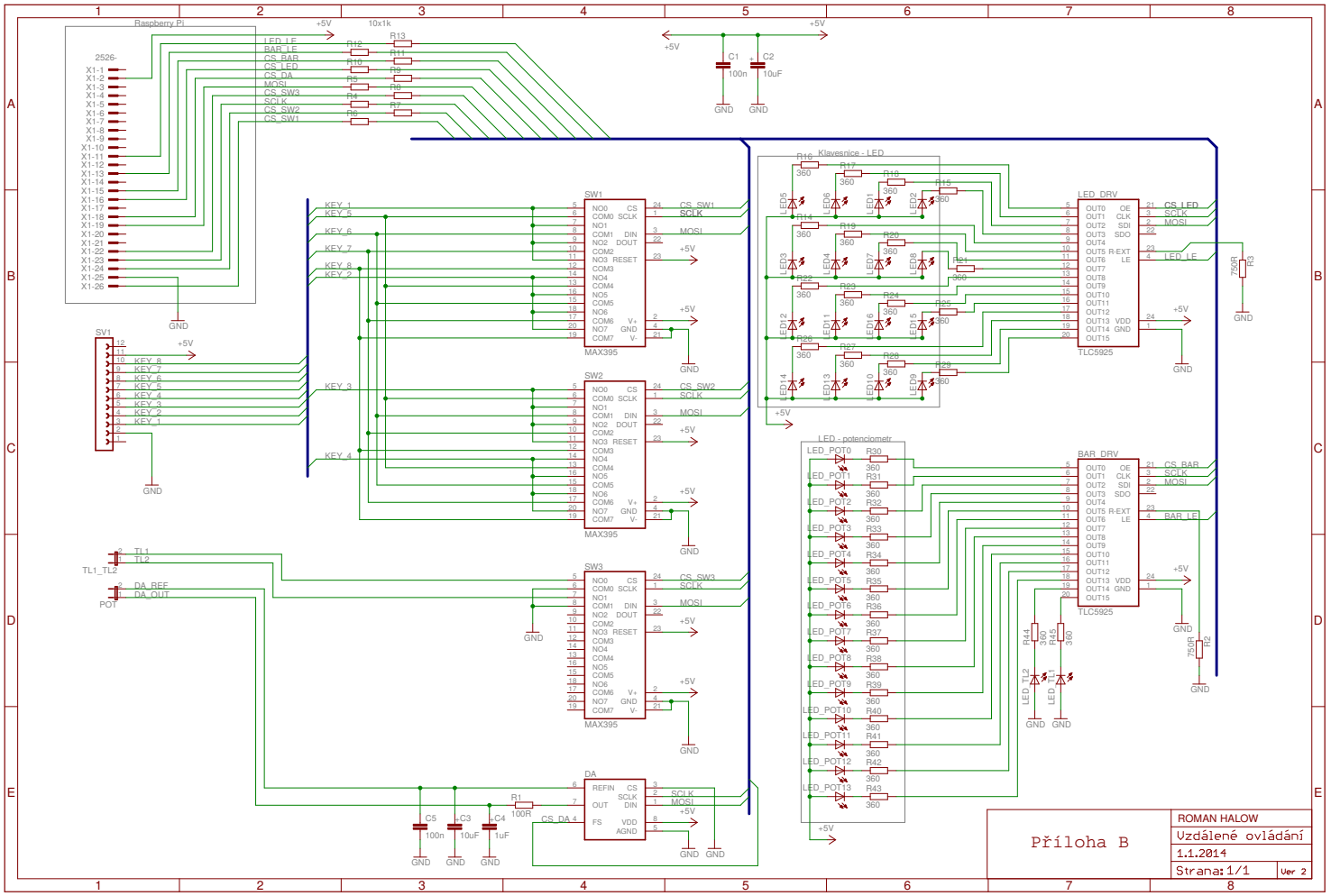
F DVD příloha

- Text diplomové práce
  - diplomova_prace_2014_Roman_Halow.pdf
  - diplomova_prace_2014_Roman_Halow_LATEX_Source.zip
- Výkresová dokumentace
  - schéma plošného spoje vzdáleného ovládání ver. 1
  - schéma plošného spoje vzdáleného ovládání ver. 2
  - schéma výukového přípravku
- Zdrojové kódy programů
  - aplikace Programmer (v programovacím jazyce JAVA)
  - aplikace SPI Controller (v programovacím jazyce JAVA)
- Katalogové listy použitých součástek
- Obraz SD karty Raspberry Pi (ve formátu *.img)
- Webové rozhraní (v archivu typu *.tar.gz)
- Zdrojový kód webového serveru Cherokee (v archivu typu *.tar.gz)
- Zkompilovaný balíček serveru Cherokee

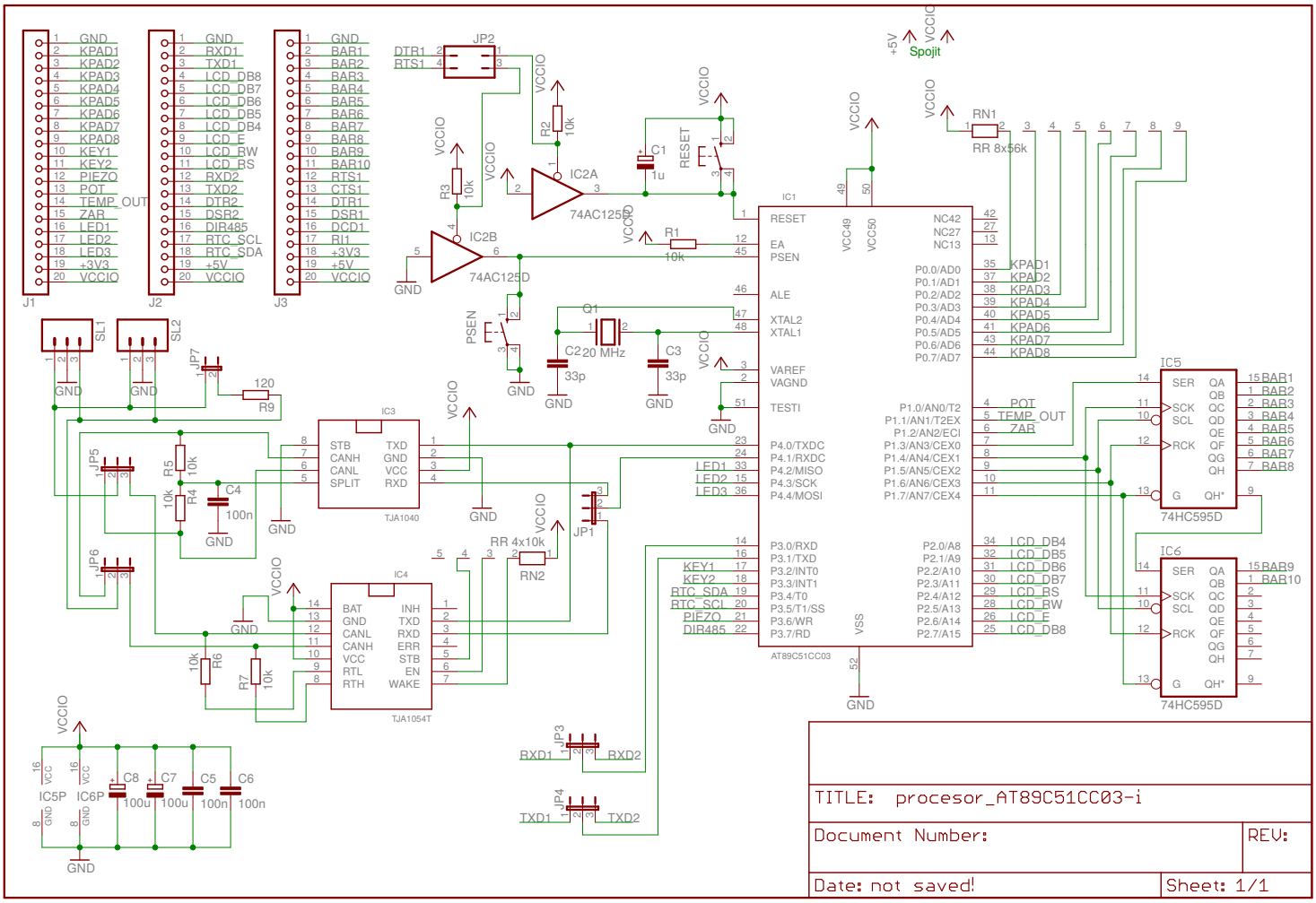
A Schéma plošného spoje vzdáleného ovládání ver. 1

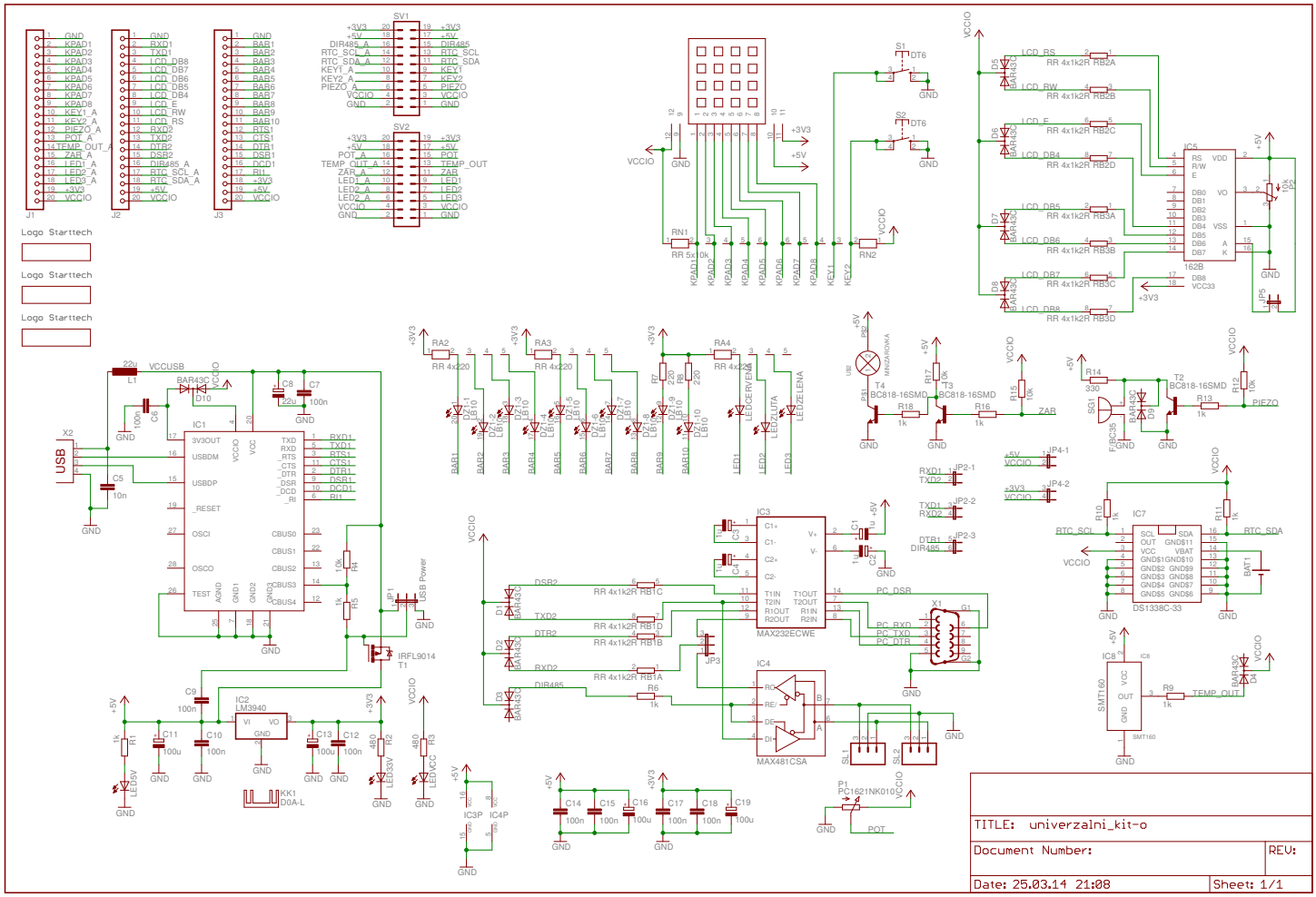


B Schéma plošného spoje vzdáleného ovládání verze 2

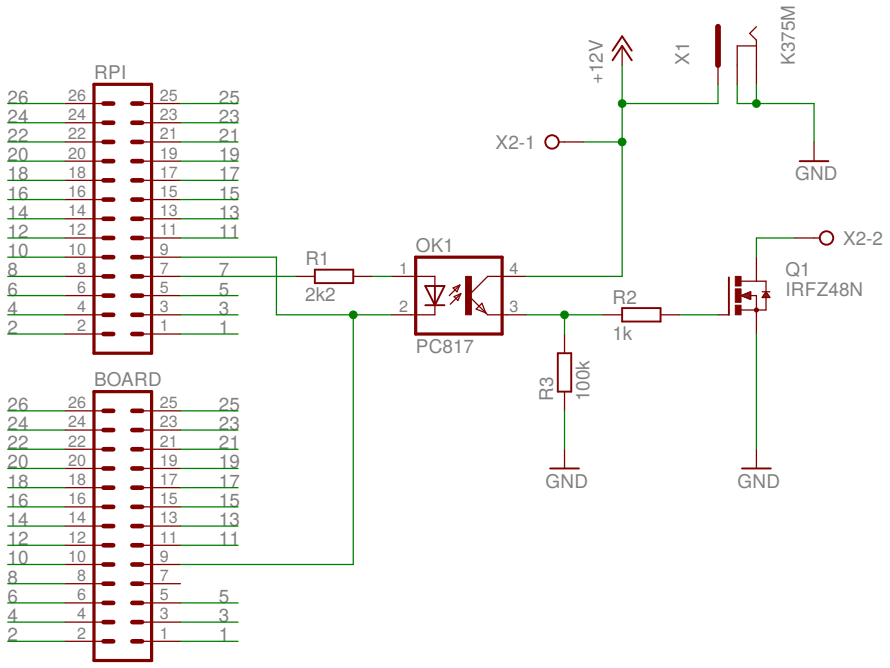


C Schéma výukového přípravku




J1
J2
J3
SV1
SV2
Logo Starttech
Logo Starttech
Logo Starttech
S1
S2
RN1
RN2
IC1
IC2
IC3
IC4
IC5
IC7
IC8
X1
X2
T1
T2
T3
T4
L1
D10
JP1
JP2
JP3
JP4
JP5
P1
RA2
RA3
RA4
USB
USB Power
IRFL9014
MAX232ECWE
MAX481CSA
DS1338C-33
SMT160
PC1621NK010
TITLE: univerzalni_kit-o
Document Number:
REV:
Date: 25.03.14 21:08
Sheet: 1/1

D Schéma ovládání osvětlení



Příloha D	Roman Halow
	osvetleni
	not saved!
	Sheet: 1/1